

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MAGYAR TUDOMÁNYOS AKADEMIA  
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZET

**PROGRAMCSOMAG ELEKTRONIKUS BERENDEZÉSEK  
HÁTLAPHUZALOZÁSÁNAK TERVEZÉSÉRE**

Irta:

KOVÁCS GYÖRGY  
FRANTA LÁSZLÓNÉ

TANULMÁNYOK 24/1974

A kiadásért felelős:

Dr. Arató Mátyás



## TARTALOMJEGYZÉK

|  |    |
|--|----|
| Bevezetés .....  | 5  |
| A programok ismertetése .....                                    | 7  |
| 1. Hátlapcsatlakozók optimális elhelyezése .....                 | 7  |
| Adatelőkészítés .....  | 8  |
| Hibaüzenetek .....   | 13 |
| Futtatási utasítás .....   | 17 |
| 2. Huzalozás minimális összhuzalhosszal .....                    | 19 |
| A program szerkezete .....                                       | 19 |
| Adatelőkészítés .....  | 21 |
| Hibaüzenetek .....   | 26 |
| Futtatási utasítás .....   | 28 |
| 3. Post—processor a TSK huzalozó félautomatához (POSTWI) .....   | 33 |
| Adatelőkészítés .....  | 33 |
| Futtatási utasítás .....   | 35 |
| 4. Post—processor a MANU—WRAP berendezéshez (MOTHER) .....       | 37 |
| Futtatási utasítás .....   | 37 |
| 5. Post—processor a TSK huzalozó félautomatához (WIRECONN) ..... | 38 |
| Adatelőkészítés .....  | 38 |
| Futtatási utasítás .....   | 42 |
| 6. Post—processor a MANU—WRAP berendezéshez (MANUWRAP) .....     | 43 |
| Adatelőkészítés .....  | 43 |
| Futtatási utasítás .....   | 46 |
| 7. A MANUWRAP2 program .....                                     | 47 |
| Futtatási utasítás .....   | 47 |
| Összefoglalás .....  | 48 |
| Irodalomjegyzék .....  | 48 |



## BEVEZETÉS

Az elektronikus berendezések hátaphuzalozását (keretkábelhuzalozás) a leggyakrabban az ún. wire-wrap technikával (rácsavart kötés) valósítják meg. A hátaphuzalozás rendkívül fárasztó és időigényes munka. Ennek a munkának a megkönnyítésére szolgál a Japánból vásárolt (TSK: SWM-1A) [1] huzalozó félautomata és az Intézetünkben kifejlesztett Manu-Wrap [2, 3] nevű berendezés. Mindkét berendezés lyukszalagvezérléssel működik és a vezérlésükhöz szükséges lyukszalagokat ún. post-processor programok állítják elő.

Jelen dolgozatunkban ismertetjük azt a programrendszert, amely a berendezések hátaphuzalozásának tervezését segíti elő és biztosítja a célautomaták vezérlését.

Az ismertetés nem terjed ki a programokban felhasznált módszerek—algoritmusok részletes leírására. Ennek oka az, hogy célunk a programokat a felhasználó szemszögéből ismertetni. Ezért leírjuk, hogyan kell a programokat futtatni, hogyan történjék az adatelőkészítés és hogy az egyes programok milyen feladatok megoldására alkalmasak.

A programrendszer programjai, az általuk megoldott feladatok szerint három csoportra oszthatók:

1. A hátlapcsatlakozók optimális elhelyezésének számítása.
2. Minimális huzalhosszal megvalósítható huzalozás számítása (minimális fák számítása).
3. Post-processor programok.

Attól függően, hogy a felhasználó mennyit bíz a tervezésből a programokra, lehetőség van a feladatot az 1., a 2. ill. a 3. alatti programokkal kezdeni.

Minden esetben meg kell adni a keret (szekrény—fiók) geometriai adatait, a csatlakozók leírását, valamint a huzalozáshoz felhasználható vezeték hosszakat. Tudni kell, hogy hány egymás feletti fiókból és fiókonként maximálisan hány csatlakozóból áll a keret. Mindezeket az adatokat, valamint azt, hogy összesen hány csatlakozót használunk a keretben és hogy egy csatlakozó egy tűskéjére hány vezeték csatlakoztatható *alapadatok*nak nevezzük.

- a) Ha az alapadatokon kívül csak a berendezés logikai rajzleírása és a tervező külön igényei állnak rendelkezésre, akkor a teljes programrendszert (1–2–3) célszerű használni. Az eredmények megadják, hogy melyik csatlakozó a keret melyik pozíciójába kerül; teljes dokumentációt kapunk a hátlapon előforduló összes jelről (szükség esetén abc szerint rendezve) és egy minimális összvezeték hosszal megvalósított huzalozás összekötendő pontpáronként megadott huzalozási listáját kapjuk, feltüntetve, hogy az egyes összekötések milyen hosszú dróttal valósíthatók meg és ezt a megadott választékból milyen kategóriájú vezetékkel lehet megvalósítani. Ezen kívül vezérlő lyukszalag kapható akár a TSK félautomata, akár a Manu-Wrap számára.



- b) Ha a felhasználó megadja nemcsak az alapadatokat és a logikai rajzleírást, hanem előre kiköti, hogy melyik csatlakozó, melyik pozícióba kerül, akkor az 1. programot csak adatkonvertálási célra használjuk. Az eredmények megegyeznek az a)-nál felsoroltakkal.
- c) Nagyon gyakran a tervezők ragaszkodnak ahhoz, hogy nemcsak a csatlakozóknak a kerten való elhelyezését, hanem az összekötési listát is maguk adják meg. Ebben az esetben csak a 3. alatti post–processor programok valamelyikének a felhasználása szükséges.

## A PROGRAMOK ISMERTETÉSE

A programok ismertetésénél bizonyos FORTRAN ismereteket és a CDC 3300 bizonyos ismeretét feltételezzük. Ezek ugyanis elengedhetetlenül szükségesek a programok használatához, viszont pl. a mostani ismertetés nem terjedhet ki a CDC 3300-as számológép ismertetésére.

### 1. HÁT LAPCSATLAKOZÓK OPTIMÁLIS ELHELYEZÉSE

#### A program lehetőségei

- a) A program maximum 20 fiókot és fiókonként legfeljebb 40 csatlakozót tartalmazó keret csatlakozóinak elrendezését tudja számítani.
- b) Összesen legfeljebb 600 csatlakozó lehet a keretben.
- c) Lehetőség van arra, hogy egyes csatlakozók helyét (fiók, pozíció a fiókban) rögzítsük a keretben (maximálisan 100 ilyen csatlakozó lehet).
- d) A program a kereten alul, felül jobbra vagy balra kívül eső csatlakozók kezelését is elvégzi. Ezek a csatlakozók mindig rögzített helyűek és más keretekkel való összekötésre szolgálhatnak.
- e) Amennyiben egyes nyomtatott áramköri lapokon nemcsak integrált áramköri elemek, hanem nagyobb alkatrészek is vannak, módunkban áll dupla (2x) ill. tripla (3x) helyet elfoglaló csatlakozókat definiálni. Ez azt jelenti, hogy az így definiált lap után egy ill. két üres hely marad a keretben.
- f) A csatlakozók csoportokba is rendezhetők. Egy csoport két vagy három csatlakozóból állhat. A csoport definíciójának értelme az, hogy az egy csoportban lévő csatlakozók mindig együtt maradnak az elrendező folyamat alatt. Egy csoportban lehet:
  - három normális (egyszeres) csatlakozó
  - két normális csatlakozó
  - egy dupla és egy normális csatlakozó
  - egy normális és egy dupla csatlakozóegymás után elhelyezve.
- g) Egy c) szerint rögzített helyű csatlakozó nem lehet f) szerinti csoport tagja.



### A program működési elve

A program Monte–Carlo módszert alkalmazva igyekszik a hátlapcsatlakozók optimális elrendezését kiszámítani. Az eljárás lényege az, hogy egy kiinduló elrendezést alapul véve (ez lehet véletlenszerű, vagy a felhasználó által meghatározott) véletlenszerűen kiválasztott csatlakozópárok cseréjével próbálja a célfüggvény értékét csökkenteni. Amennyiben a csere után a célfüggvény nem csökken, a pár elemeit visszateszi eredeti helyükre és megjegyzi, hogy ez a csere sikertelen volt. A sikertelen cserékben részt vevő párok tárolása minden sikeres csere után újakezdődik. Így a program futása során egyre nehezebben fog olyan párt kiválasztani, amelyek cseréje a célfüggvény csökkenését eredményezné. Egy optimum (lokális optimum!) megtalálását az egymás utáni sikertelen cserekísérletek regisztrálásával tudjuk biztosítani. Ha  $N$  csatlakozó esetén  $K=N(N-1)/2$  egymásutáni különböző sikertelen párcserékísérletet hajtunk végre, akkor biztos, hogy az elrendezés párcserékkel már nem javítható. A mi esetünkben általában nincs szükség  $K$  sikertelen cserekísérletre, egyrészt, mivel a csoportok és a rögzített helyű csatlakozók miatt a lehetséges sikertelen párcserek száma  $< K$ , másrészt, ha valamivel kisebb számú sikertelen párcsere kísérlet után állítjuk le a folyamatot, akkor is megközelítjük a lokális optimumot a futásidő jelentős lecsökkenése mellett.

A program nemcsak egy, hanem több lokális optimum kiszámítására alkalmas, úgy, hogy mindig új véletlenszerű elrendezésből indul (kivéve az első elrendezést, ami lehet a felhasználó által megadott is). Így több eredmény–elrendezés állhat rendelkezésünkre, amelyek közül választási lehetőségünk van. Minél több lokális optimumot számolunk, annál nagyobb a valószínűsége az abszolút optimum elérésének, viszont természetesen annál drágább a futás.

Célfüggvényként egy, a tényleges vezetékhozzal arányos hosszt számolunk. A számítás úgy történik, hogy minden egyes jelre (összekötendő pontcsoport) nézve kiszámítjuk a jel által érintett pontok teljes gráfjának összhosszát és ezeket a hosszokat összegezzük. Egyszerűsítésként nem pythagorasi, hanem derékszögű hosszakkal számolunk és azt is feltételezzük, mintha egy csatlakozó minden kivezetése annak közepéből jönne ki. Így a sokszorosan összekötött elemekre nézve az összekötöttség mértékét mesterségesen megnöveljük és a számítás sokkal gyorsabb, mintha valódi összvezetékhozzát számolnánk.

### Adatelőkészítés

A program bemenő adatai a keret geometriai leírását, a futtatás paramétereit és a keretkábelen összekötendő pontok összekötési listáit tartalmazzák jellista formájában. A keretben szereplő csatlakozók azonosítására vagy 1-től kezdődő egész számok, vagy alfanumerikus karaktersorozatok szolgálnak. Az egyes jelek vagy forráspontjuk azonosítóival (csatlakozó azonosító és tüske szám) vagy alfanumerikus nevekkal adhatók meg. A programfutás fő paramétereit egy vezérlő

kártya szabályozza. Ez a következő információkat tartalmazza (változó nevekkkel megadva):

|        |  |
|--------|--|
| LAB(I) | A keret azonosítási neve, amely max. 6, két egymás utáni space-t nem tartalmazó karakterből állhat.  |
| JAVSZ  | Hány lokális optimumot kívánunk számítani.   |
| MUST   | Egy lokális optimum kiszámításának jóságát befolyásoló változó. Azon egymás utáni sikertelen párcserekísérletek számát adja meg, amely után az eljárást abbahagyhatjuk. $k$ elem esetén $MUST=k(k-1)/2$ garantálja a lokális optimum megtalálását, tehát ennél nagyobb szám megadása értelmetlen, ennél kisebb szám megadása esetén a programfutás gyorsabb (tehát olcsóbb) lesz, ennek ára viszont egy esetleg rosszabb eredmény. |
| INMOD  | A csatlakozók kezdeti elrendezésének megválasztási módját határozza meg. Lehet:<br>666 – Random kezdeti elrendezés.<br>0 – A csatlakozókat azonosító egész számok szerint sorfolytonosan helyezi el a csatlakozókat a program a keret szabad helyein.<br>100 – Adatként kell megadni az induló elrendezést.  |
| LAPSZ  | A kereten lévő csatlakozók száma (a kereten kívül lévő csatlakozókat is beleértve).  |
| ICS    | A kereten kívül elhelyezett, rögzített csatlakozók száma.  |
| M      | A keret sorainak (fiók) száma.   |
| N      | Csatlakozók száma fiókonként.  |
| KX     | Két szomszédos csatlakozó (oszlopok) közti távolság.   |
| KY     | Két szomszédos fiók (sorok) közti távolság.  |
| KARAK  | Megadja, hogy a bemenő adatok csak számokból vagy számokból és karakter sorozatokból állnak-e. Lehet:<br>0 – Egész számok.<br>100 – Karakter sorozatok és egész számok.  |
| MODUL  | Az egy modulba elhelyezett csatlakozók számát adja meg.  |

#### MEGJEGYZÉSEK

- Ha nincs modulosztás a fiókokban, akkor  $MODUL=N+2$  értéket kell megadni.
- A KX és KY értékei nem valódi távolságok. A program szempontjából csak a KX/KY arányának van jelentősége. Ennek reális értékei  $1/2$  és  $1/10$  közé esnek.
- N és M értékébe csak a kereten belüli sorok ill. oszlopok száma számít bele, a kereten kívüli csatlakozók sor ill. oszlopszáma rendre  $0, N+1$  ill.  $0, M+1$  lehet.



- Ha KARAK=0, a csatlakozók azonosítói a természetes egész számok: 1, 2, ..., LAPSZ, amíg az első ICS név (1, 2, ..., ICS) a kereten kívüli csatlakozókat jelenti.
- Ha KARAK=100, akkor a csatlakozók azonosítói 16 alfanumerikus karakternél nem hosszabb, két egymás utáni space-t nem tartalmazó karaktersorozatok.
- Ha KARAK=100, minden jelnek egy 54 alfanumerikus karakternél nem hosszabb, két egymás utáni space-t nem tartalmazó azonosítója kell, hogy legyen.
- Ha KARAK=0, akkor a jel–forráspont csatlakozó és túske számából generál a program egy jelnevet, amely azonosításra alkalmas.
- Ha KARAK=100, akkor a csatlakozók max. megengedett száma 300 (a 600 helyett), a csatlakozók azonosítói átlag 7 karakterből állhatnak; a megadható jelek maximális száma 600, átlagos hosszuk 7 karakter lehet.
- Az összes bemenő adatot a később megadandó formátumok szerint lyukkártyára kell lyukasztani.

A következőkben ismertetjük, hogy milyen bemenő adat, milyen célt szolgál és milyen formátum szerint kell lelyukasztani. A KARAK=100 (karakteres forma) szerinti adatokat a-val a KARAK=0 (egész számok) szerinti adatokat b-vel jelöljük. Egy programfutás minden adatát vagy a vagy b szerint kell előkészíteni. Csak az első adatkártya, a vezérlő kártya az, amelyik közös mindkét esetben. A bemenő adatok csoportokat képeznek, amelyeket üres kártyák választanak el egymástól. Ha bármely csoport nem szerepel az adatok között (pl. nincs rögzített helyű csatlakozó), az üres kártyát akkor is be kell tenni az adatok közé. A vezérlő kártya után nincs üres kártya.

#### Adatcsoportok:

##### *1. a. és b. Vezérlő kártya*

Formátum: (6A1,1x,11I4)

Tartalmát már ismertettük. Rendre a következő változókat tartalmazza: (LAB(I), I=1,6), JAVSZ,MUST,INMOD,LAPSZ,ICS,M,N,KX,KY,KARAK,MODUL.

##### *2. A rögzített helyű, valamint a dupla és tripla csatlakozók definíciója*

a. Formátum: (16A1,3I4)

Tartalom: (EXT(I), I=1,16),ME,IS,IOSZ

b. Formátum: (2X,4I4)

Tartalom: NEV,ME,IS,IOSZ

A különbség a két formátum között az, hogy a konnektor neve az a esetben egy max. 16 karakterből álló sorozat (EXT(I)), míg a b esetben egy egész szám (NEV). ME adja meg a csatlakozó méretét. Ez 0,1,2 vagy 3 lehet (a 0 ugyanazt jelenti, mint az 1), ami szimpla, dupla, vagy tripla helyigényű csatlakozót jelent.

Ha a csatlakozó helye nem rögzített, akkor IS és IOSZ értéke nulla kell, hogy legyen, míg, ha az adott csatlakozó helye rögzített, akkor IS adja meg a fiók számát és IOSZ a csatlakozó helyét a fiókon belül (sor- ill. oszlopszám).

### 3. Csoport definíciók

a. Formátum: (3(16A1,4X))

Tartalom: (KOT(J,I),I=1,16),J=1,3

b. Formátum: (2X,3I4)

Tartalom: IG1(NG),IG2(NG),IG3(NG)

Minden egyes adatkártya egy csoport 2 ill. 3 összetartozó tagját adja meg. Ha egy csoportnak csak két tagja van, a harmadik hely üresen marad. Az a esetben a nevek a KOT(I,J) kétdimenziós tömbbe kerülnek, majd feldolgozás utáni belső azonosítók a b-hez hasonló három egydimenziós tömbbe kerülnek, ahol NG jelöli, hogy hányadik csoportról van szó és IG1 az első IG2 a második és IG3 a harmadik csoporttag azonosítóját tartalmazza. Természetesen, ha a k-adik csoportnak csak két eleme van, akkor  $IG3(k)=0$ .

### 4. Összekötési lista

a. Formátum:

1. első kártya: (54A1,6X,16A1,I1,I3)

2. folytatás kártya: (4(16A1,I1,I3))

Tartalom:

1. (JELNEV(I),I=1,54),(IFOR(I),I=1,16),IVEG,JEL2

2. ((KOT(J,I),I=1,16),IFI(J),IPPI(J)),J=1,4

Egy összekötések nélküli pont megadható egy kártyán, míg több pont összekötése egy 1. kártyán és tetszőleges számú folytatás kártyán (2.) adható meg. Az egy jelhez tartozó összes összeköttetést, csakúgy, mint a b esetben, együtt kell megadni. Az egyes változók értelmezése:

|        |   |
|--------|---|
| JELNEV | tartalmazza a jel nevét   |
| IFOR   | a csatlakozó neve   |
| JEL2   | az IFOR nevű csatlakozó melyik tűskéjéről (pin) indul a jel (forrás)  |
| IVEG   | ha ennek értéke 1, akkor nincs folytatás kártya   |
| KOT    | a folytatás kártyán beolvasott csatlakozónevek  |
| IPPI   | a KOT-ben lévő csatlakozókhoz tartozó tűskeszámok   |
| IFI    | ha ezen tömb bármely elemének értéke 1, azt jelenti, hogy nincs több folytatás sor, tehát vagy egy 1. kártya, vagy ha a lista utolsó jele volt, akkor üres kártya következik. |



b. Formátum: (2X,18I4)

Tartalom: JEL1,JEL2,(LAP(I),PIN(I),I= 1,8)

|      |  |
|------|--|
| JEL1 | a jelforrás csatlakozójának neve (száma)         |
| JEL2 | a jelforrás tűske száma a JEL1 csatlakozón       |
| LAP  | az összekötendő pontok csatlakozó nevei (számai) |
| PIN  | a LAP-nak megfelelő tűske számok.                |

Ha a jel nem érint több mint 9 pontot, akkor egy kártyán megadható, ha több, mint 9 pontot kell összekötni, akkor folytatás kártyára (kártyákra) van szükség. A folytatás kártyán a JEL1 és JEL2 változóban meg kell ismételni a jelforrás adatait. Pl. egy folytatás kártyával maximum  $9+8=17$  összekötendő pont adható meg.

Látható, hogy mind az a, mind a b esetben az összekötendő pontok csatlakozó és tűske (pin) azonosítóit kell megadni úgy, hogy egy jelhez tartozó összes összekötendő pont együtt szerepeljen.

5. Adott induló elrendezés

a. Formátum: (16(A1,2I4))

Tartalom (LAPNEV(I),I= 1,16),ISOR,IOSZL

b. Formátum: (2X,3I4)

Tartalom: NEV,ISOR,IOSZL

Az induló elrendezés LAPSZ számú adatkártyát kell, hogy tartalmazzon abban az esetben, ha a vezérlő kártyán INMOD=100. Ha nincs adott kiinduló elrendezés, akkor az egyébként utána kötelező üres kártyát nem szabad az adatok közé tenni. Az a és b között csak annyi a különbség, hogy az a esetben a csatlakozó neve egy karaktersorozat (LAPNEV), míg a b esetben egy egész szám (NEV). Az ISOR és IOSZL változók az egyes csatlakozók helyét adják meg (ISOR=fiókszám, IOSZL=sorszám a fiókban).

6. a. és b. Új feladatkártya

Formátum: (2X,I4)

Tartalom: ITOV

Az utolsó adatcsoport és az azt követő üres kártya után következő kártyán lévő adat megadja, hogy van-e még adat. Ha ITOV=0, akkor nincs több megoldandó feladat, ha ITOV $\neq$ 0, akkor a következő adatkártya a következő feladat vezérlő kártyája kell hogy legyen.

7. a. és b. Program adat vége

Ha nincs több megoldandó feladat, akkor a 6. szerinti ITOV=0 és ez után egy olyan kártyával zárjuk az adatsomagot, amelynek az első 30 pozícióján a 9-es számot kell lelyukasztani.



## MEGJEGYZÉSEK

- Ha a programfutás sikeres, akkor minden feladatra vonatkozóan megkapjuk az összes kiinduló és lokális optimum elrendezést a hozzájuk tartozó célfüggvény–értékekkel együtt.
- A legjobb célfüggvény–értékkel rendelkező elrendezés adataival minden feladat számára a "huzalozó program" bemenő adatait készíti el a program a dsi=41-es diszk file-ra.
- Sikeres programfutás után a program STOP 02 üzenettel áll le és "end of file" jellel lezárja a 41-es dsi-jű diszk file-t.
- Ha az adatok közt vagy futás közben hibát észlel a program, akkor hibaüzenetet ad és STOP 7 vagy STOP 77-tel leáll és a futás befejeződik.

### Hibaüzenetek

### Jelentés

#### INPUT DATA ERROR

Számos esetben használjuk:

- A sorok vagy oszlopok száma nagyobb a megengedettnél.
- A csatlakozók száma nagyobb a megengedettnél.
- Csatlakozó mérete nem megengedett (nem 1,2 vagy 3).
- Helytelenül rögzített rögzített helyű csatlakozó.
- Csoport elemeinek száma nagyobb, mint három.
- Az összekötési listán megadott túl nagy azonosítójú csatlakozó.
- Több rögzített helyű csatlakozó, mint amennyit a vezérlő kártya megad.
- A karakter formában megadott csatlakozók száma nagyobb a vezérlőkártyán megadottnál.

#### ERROR: TOO MUCH INPUT DATA

Az összekötési lista túl sok adatból áll, a program nem futhat.

#### ERROR INFIX

Az adott induló elrendezés adatkártyáin fellépő hiba.

#### ERROR WRIFIX

Helytelen helymegadás rögzített helyű csatlakozó számára.

| Hibaüzenetek | Jelentés  |
|--------------|---|
| ERROR FILL   | Nincs elegendő szabad hely az összes csatlakozó elhelyezésére, ha nem adott az induló elrendezés.<br>(INMOD≠100 esetén) |
| ERROR FILFIX | A program nem képes a szabad csatlakozóhelyek feltöltésére adott induló elrendezés esetén (INMOD=100).                  |
| ERROR KODLAP | A csatlakozók listája túl sok elemből áll, vagy a csatlakozók száma nagyobb a megengedettnél.                           |
| ERROR KODJEL | A jeleket tartalmazó lista túl sok elemből áll, vagy a jelek száma nagyobb a megengedettnél.                            |

## MEGJEGYZÉS

Az egyes hibaüzenetek hatására a programfutás abbamarad és új futtatásra csak a hiba kiküszöbölése (pl. adatkártya javítás) után van mód. Az adathibák kiszűrését elősegíti, hogy a program az összes adatot még feldolgozás előtt kilistázza a szélesnyomtatón és így, mivel minden adat formázottan kerül a gépbe mód van egy jól áttekinthető lista szemmel történő formai ellenőrzésére. Ezt az ellenőrzést meg lehet csinálni az adatok gépbe vitele előtti off-line lista alapján is.

Természetesen az adatokat célszerű többször ellenőrizni, mivel az összekötési lista olyan hibáit, ahol egy helyes szám helyett az értékhatárokon belüli helytelen szám kerül a program nem képes észrevenni. Ez a veszély nem áll fenn, ha a program adatait más programok eredményei szolgáltatják.

## A program szubrutinjai

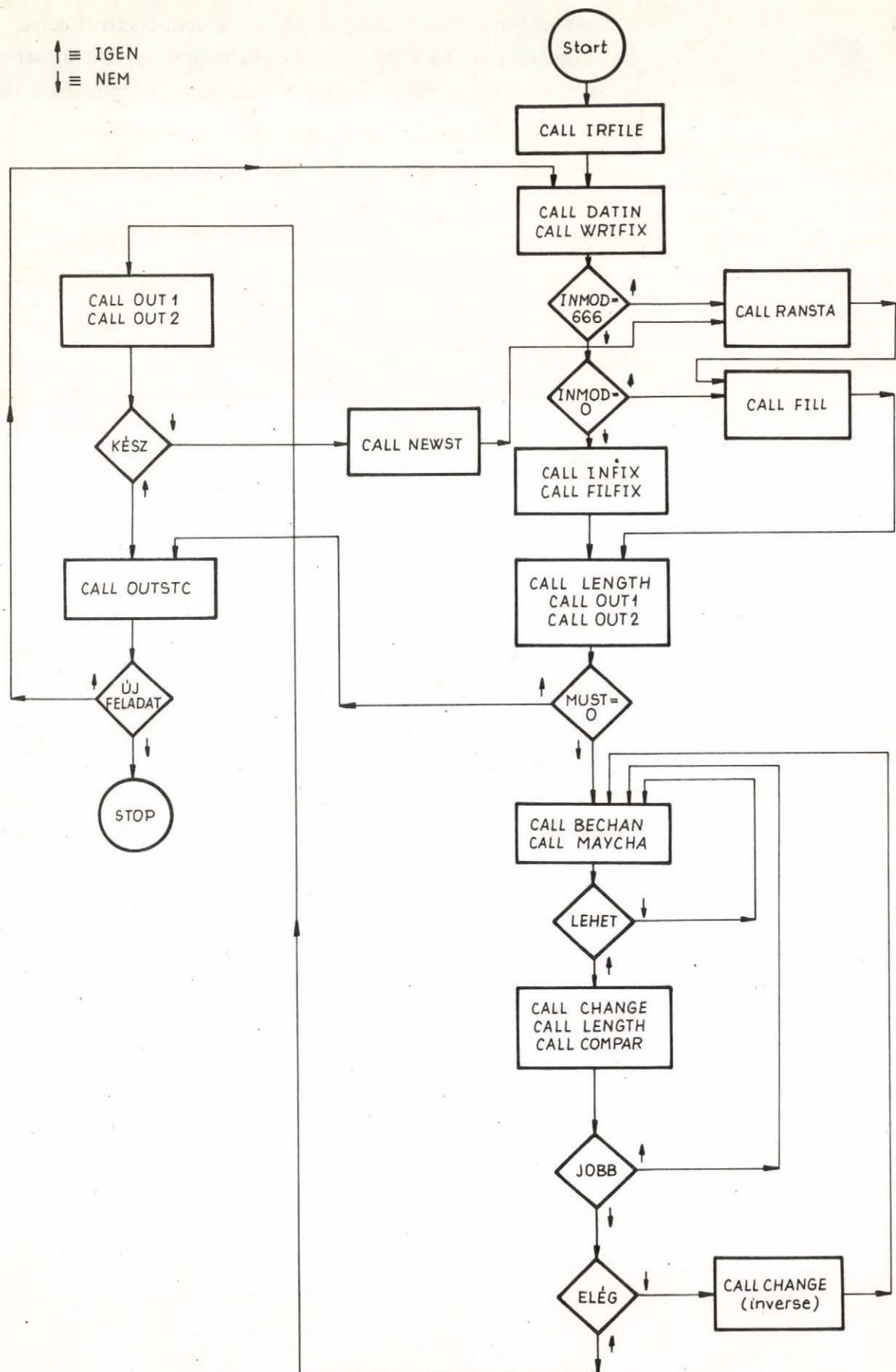
|        |   |
|--------|---|
| IRFILE | Az összes adatot beolvassa lyukkártyáról (DSI=7) és egyrészt egy diszk file-ra (DSI=6), másrészt az LP-re (DSI=5) kiírja. |
| DATIN  | Beolvassa az adatokat a DSI=6 diszk file-ról és további feldolgozásra alkalmas formában eltárolja a központi memóriában.  |
| INFIX  | Az adott induló elrendezés (ha INMOD=100) adatait olvassa be.   |
| WRIFIX | Helyükre teszi a rögzített helyű csatlakozókat.   |



|         |   |
|---------|---|
| FILL    | A keret szabad csatlakozóhelyeit tölti fel a nem rögzített helyű csatlakozókkal ill., ha a csatlakozóhelyek száma nagyobb a csatlakozók számánál, a maradék helyekre pszeudo-csatlakozókat rak. |
| RANSTA  | Véletlenszerű csatlakozó elhelyezést biztosít a kereten.  |
| BECHAN  | Az optimalizáló folyamat számára választja ki a két felcserélendő csatlakozót.  |
| CHANGE  | Elvégzi a két kiválasztott csatlakozó helyének felcserélését.   |
| GROUP1  | Megvizsgálja, hogy egy csatlakozó egy definiált csoport első eleme-e?   |
| GROUP23 | Megvizsgálja, hogy egy csatlakozó egy definiált csoport második vagy harmadik eleme, vagy sem.  |
| MAYCHA  | Megvizsgálja, hogy a BECHAN rutin által kiválasztott csatlakozók helyét fel lehet-e cserélni.   |
| COMPAR  | Összehasonlítja a csere előtti és utáni célfüggvény értékeit.   |
| LENGTH  | A célfüggvény számítása.  |
| FILFIX  | Adott induló elrendezés esetén a keret szabad csatlakozóhelyeinek feltöltését végzi.  |
| KODLAP  | Eltárolja a csatlakozóneveket azonosító karaktersorozatokat (ha KARAK=100), egy egész számot rendel minden névhez és eredményközléskor visszakeresi a számhoz tartozó nevet.                    |
| KODJEL  | Ugyanazt végzi, mint a KODLAP rutin, de nem a csatlakozó-, hanem a jelnevekkel.   |
| RANDA   | Ez egy REAL FUNCTION, amely a CDC könyvtári RANDU véletlenszám-generátort használva generál véletlenszámokat $0 < x < 1$ .  |
| OUTSTC  | A programfutás eredményét egy diszk file-ra viszi (DSI=41), hogy a huzalozó program (CONEX) számára adatként legyen használható.  |
| OUT1    | Az elrendezés kinyomtatását végzi. Csatlakozók elhelyezése és csatlakozónév-azonosító szám hozzárendelés kiírása (ha van ilyen).  |
| OUT2    | Az elrendezés mátrix formában történő kinyomtatását végzi.  |
| NEWST   | Uj optimum kereséséhez végez előkészítést (nullázás stb.).  |

#### MEGJEGYZÉSEK

- A szubrutinok kapcsolódását és működési sorrendjét a mellékelt blokkdiagram ábrázolja (1. ábra).
- A program nyelve FORTRAN IV.



1. ábra

- Memória igény: 80 negyed lap (40k szó x 24 bit).
  - Szükséges perifériák:
    - Kártyaolvasó (dsi=7)
    - Sornyomtató (LP) (dsi=5)
  - 8. diszk egység:
    - Adat file (dsi=6)
    - Eredmény file (dsi=41)
    - A lefordított program (dsi=34)
  - A futásidő a csatlakozók számától (LAPSZ), a számítandó lokális optimumok számától (JAVSZ) és a MUST változó értékétől függ.
- Egy 24 csatlakozót tartalmazó keret egy lokális optimumának kiszámítása kb. 2–5 percet igényel a CDC 3300-on.

### Futtatási utasítás

A programfutáshoz a 8. diszk egységre és scratch file-ra van szükség.

A program egy \*DEF állandó file-ra van lefordítva (dsi=34) és így a következő struktúrával futtatható:

```
$JOB, szám, név, idő, sorok száma,, , megjegyzés
$SCHED,CORE=80,CLASS=B,SCR=30,854=1
$FILE,7=INP
$FILE,5=OUT
$*DEF(O,W,34,222025,KERT,1,GYR,O)
$*DEF(O,W,41,222025,KGY4,1,GYR,O)
$X,34
```

Adatkártyák

EOJ kártya

- Ha a program FORTRAN dekkjét akarjuk használni, úgy, hogy módosítsuk a programot és a módosított programot akarjuk a dsi=34 file-on megtartani, akkor a  
\$X,34 kártya helyett a következőket kell használni

```
.
.
.
$FTNU(X=34,L,S,D)
Programkártyák
$X,34
.
.
.
```



- Ha a programot nem akarjuk lefordítani a dsi=34 file-ra, akkor nincs szükség a

\$\*DEF(O,W,34,...) kártyára és

\$X,34 kártya helyett a következőket használjuk:

.

.

\$FTNU(X,L,S,D)

Programkártyák

\$X,LGO

.

.

- A program természetesen a perifériák máshogy történő felhasználása esetén is futtatható, pl. a dsi=6-os munka—file helyett SOCR vagy \*DEF file is használható, vagy a \*DEF(O,W,41,...) file helyett is használható 41-es dsi-jű munka- vagy SOCR file. (Ez utóbbinak akkor van értelme, ha az elhelyező programot egy job-ban futtatjuk a huzalozó programmal. Egy ilyen futás dekkjének összeállítását a CONEX program ismertetése után adjuk meg.)

## 2. HUZALOZÁS MEGVALÓSÍTÁSA MINIMÁLIS ÖSSZHUZALHOSSZAL

A program a hátlapcsatlakozókon kijelölt  $N$  pont összekötését biztosítja az összekötések geometriai hosszának minimalizálása mellett. A program minimális fákkal megvalósított összekötéseket számol figyelembe véve, hogy megadható egy korlátozás az egy ponton megengedett kötések számára.

### A program működési elve

A program a [4]-ben ajánlott stratégia alapján dolgozik. A gyakorlati megvalósításban (2. ábra) némileg eltértünk az eredeti algoritmustól és figyelembe vettük a fent említett korlátozást. Az egy pontra megengedett kötések száma minden egyes tűskére (pin) nézve ugyanaz.

### A program szerkezete

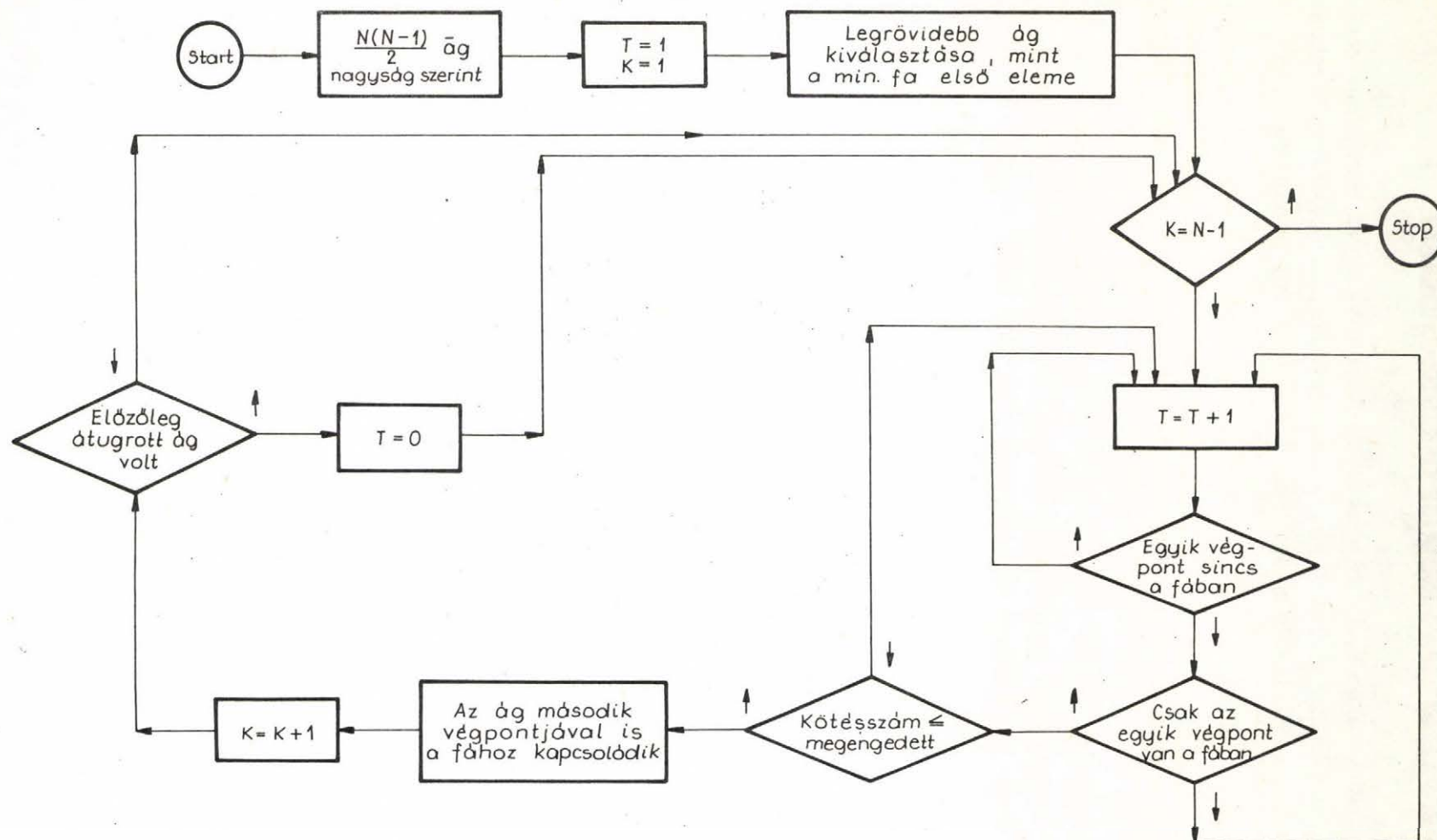
A program futása előtt egy adatrendező–előkészítő program futtatása szükséges. Ez a program beolvassa a lyukkártyán előkészített adatokat ( $dsi=7$ ) és átteszi egy diszk file-ra ( $dsi=20$ ), majd ha az elhelyező program eredményét felhasználva futtatjuk (általában ez történik), akkor az előző futás eredményét beolvassa egy diszk file-ról ( $dsi=41$ ) és szintén átteszi a  $dsi=20$  jelű file-ra. Így a huzalozó program már erről a file-ról veszi az összes bemenő adatát.

A huzalozó program bemeneti, számítást végző és kimeneti részből áll. A bemenetet végző rész először beolvassa és feldolgozza a geometriai alapadatokat. Ezután a program mindhárom része már minden összekötendő pontcsoportra külön–külön számítja a minimális fákat. Azt is mondhatjuk, hogy a programfutás minden csoportra (minden jelre) nézve újra kezdődik. Ahhoz, hogy így működhessen a program, biztosítanunk kell, hogy az egy csoporthoz tartozó összes pont együtt (egymás után) szerepeljen az adatok között. Ezt vagy az adatok megfelelő összeállításával, vagy egy előzetes rendezéssel érhetjük el, vagy így kapjuk, ha adataink korábbi programok eredményei.

*Input (bemenet) rész.* Ez a programrész a geometriai alapadatok feldolgozása után egy–egy összekötendő pontcsoport beolvasását végzi.

*Számítást végző rész.* A keret és a csatlakozók geometriai adatai felhasználásával a beolvasott pontcsoport pontjainak helyét, majd az  $N$  pont által meghatározott  $N(N-1)/2$  távolságot számítja ki. Ezeket nagyság szerint rendezi, majd az összekötéseket megvalósító  $N-1$  ágot választja ki. Az összekötések számítását minimális összvezeték-hossz felhasználásával végzi, figyelembe véve az egy pontra tehető kötőszám korlátozást (lásd a 2. ábrát).

2. ábra



N = pontok száma

T = ág nagyság szerinti sorszáma

K = a fába beépített ágak száma

↑ ≡ Igen

↓ ≡ Nem



*Output (kimenet) rész.* Ez a következő eredményeket állítja elő:

- A sornyomtatón kapunk egy jellistát, ahol minden egyes jel után felsorolja az összes geometriai helyet, ahol a jelnek jelen kell lennie. Ha használjuk a SORT rutint, akkor ez a lista abc sorrendben van.
- Az összekötendő pontpárok jelenkénti listája a sornyomtatón és a diszk file-on is megjelenik, megadva minden jelhez a minimális fát alkotó összekötendő pontpárokat, az összekötést megvalósító vezeték hosszát és az ehhez alkalmas huzalkategóriát. A file-ra jutó adatok a huzalozó automaták vezérlő információját előállító post-processor programok adatait képezik.
- Egy táblázatot kapunk a sornyomtatón, amely megadja, hogy melyik huzalkategória milyen hosszú huzalt jelent és abból hány db-ra van szükség a huzalozáshoz.
- Kiírja a program azt is, hogy a teljes huzalozáshoz mennyi huzalra van szükség.

### **Adatelőkészítés**

A program bemenő adatai 80 oszlopos lyukkártyán vagy hasonló formátumban diszk vagy mágnesszalag tárolón lehetnek, vagy ezek közül több is használható. Természetesen eszerint az adatelőkészítő program módosítása válhat szükségessé. Négyféle adattípust különböztetünk meg, amelyek közül az 1. a teljes futásra, a 2. és 3. egy vagy több feladat megoldására, a 4. pedig csak egy feladatra érvényes. Egy feladat alatt egy huzalozandó keretet értünk. A 2. és 3. típusú adat minden feladatra megadható, de ha csak az elsőre adjuk meg, akkor érvényes marad mindaddig, amíg nem adunk még újabb, abba a típusba tartozó adatot.

#### 1. típus. Szabványos vezetékhozzak (kategóriák)

| Pozíció | Hossz | Adat  | Tipus       |
|---------|-------|-------|-------------|
| 1–18    | 18    | blank | konstans    |
| 19–20   | 2     | LG    | konstans    |
| 21–65   | 46    | blank | konstans    |
| 66–70   | 5     | hossz | egész x1 mm |
| 71–80   | 10    | blank | konstans    |

## 2. típus. Geometriai alapadatok

| Pozíció | Hossz | Adat                             | Típus         |
|---------|-------|----------------------------------|---------------|
| 1–6     | 6     | feladatnév                       | alfanumerikus |
| 7–18    | 12    | blank                            | konstans      |
| 19–20   | 2     | PA                               | konstans      |
| 21–22   | 2     | blank                            | konstans      |
| 23–29   | 7     | sortávolság<br>(y irány)         | egész x0.1 mm |
| 30–31   | 2     | blank                            | konstans      |
| 32–38   | 7     | modultávolság<br>(x irány)       | egész x0.1 mm |
| 39–40   | 2     | blank                            | konstans      |
| 41–47   | 7     | csatlakozótávolság (x irány)     | egész x0.1 mm |
| 48–49   | 2     | blank                            | konstans      |
| 50–56   | 7     | megengedett kötésszám egy ponton | egész x0.1 mm |
| 57–80   | 24    | blank                            | konstans      |

## 3. típus. Csatlakozó tűske térkép

| Pozíció | Hossz | Adat                | Típus         |
|---------|-------|---------------------|---------------|
| 1–6     | 6     | feladatnév          | alfanumerikus |
| 7–18    | 12    | blank               | konstans      |
| 19–20   | 2     | PI                  | konstans      |
| 21–22   | 2     | blank               | konstans      |
| 23–29   | 7     | x koordináta        | egész x0.1 mm |
| 30–31   | 2     | blank               | konstans      |
| 32–38   | 7     | y koordináta        | egész x0.1 mm |
| 39–65   | 27    | blank               | konstans      |
| 66–70   | 5     | pinszám (azonosító) | egész         |
| 71–80   | 10    | blank               | konstans      |



#### 4. típus. Összekötendő pontok listája

| Pozíció | Hossz | Adat                | Tipus         |
|---------|-------|---------------------|---------------|
| 1–6     | 6     | feladatnév          | alfanumerikus |
| 7–60    | 54    | jelnév              | alfanumerikus |
| 61      | 1     | blank               | konstans      |
| 62      | 1     | F                   | konstans      |
| 63–64   | 2     | keretszám           | egész         |
| 65      | 1     | blank               | konstans      |
| 66      | 1     | R                   | konstans      |
| 67–68   | 2     | fiókszám            | egész         |
| 69      | 1     | blank               | konstans      |
| 70      | 1     | M                   | konstans      |
| 71–72   | 2     | modulszám           | egész         |
| 73      | 1     | blank               | konstans      |
| 74      | 1     | K                   | konstans      |
| 75–76   | 2     | csatlakozó-<br>szám | egész         |
| 77      | 1     | blank               | konstans      |
| 78      | 1     | P                   | konstans      |
| 79–80   | 2     | tüskeszám           | egész         |

#### MEGJEGYZÉSEK

- A fiók, modul, csatlakozó és tüske azonosítók egész számok 0-tól kezdve.
  - A legnagyobb megengedett tüskeszám: 100.
  - Az 1. típusú adat a huzalozó automaták huzaltároló egységeibe kerülő huzalhosszakat határozza meg, a hozzájuk rendelt azonosító számok (0–99) megfelelnek a hosszak megadási sorrendjének.
  - Legfeljebb 100 különböző huzalkategória megadása lehetséges.
  - A 2. típusú adat egyetlen rekord (adatkártya), amely a keretre vonatkozó általános információkat tartalmazza.
  - A 3. típusú adat egy csatlakozó összes tüskéjének azonosítószámát és az adott tüske egy tetszőleges nulla ponttól vett x és y távolságát adja meg.
- Természetesen ez a nulla pont közös az összes tüskére nézve.
- Egy keret összes csatlakozójának azonos tüske–térképe kell, hogy legyen.

- A 4. típusú adat az összes felhasznált tuskét megadja a jelnévvel és a jelhez tartozó tuskék geometriai helyével (keret, fiók, modul, csatlakozó túske).
- Egy jelhez max. 20 túske tartozhat. Tehát, ha egy jel több, mint húsz pontot érint, azt több részre kell bontani.
- A program az összes 4. típusba tartozó adatot (ha van rendezés, akkor azután) kiírja a sornyomtatón, oly módon, hogy egy oldalon nem ír 50 sornál többet, viszont az egy jelhez tartozó összes pont ugyanarra az oldalra kerül. A feladatnév, a jelnév és a keretszám minden jelnél csak egyszer, az adott jel első pontjához lesz kiírva.
- Eredményként megkapjuk az egy jelhez tartozó összeköttetéspárok felsorolását, mindig megjelölve, hogy a jel hány pontot érint. Az összekötendő párok egy sorba kerülnek. Ugyanebben a sorban van a két pont távolsága és a megvalósításhoz szükséges huzalkategória is. A pontok keret, fiók csatlakozó és tuskeshám koordinátákkal szerepelnek. Ez a lista alkalmas ellenőrzési célra és szükség esetén ennek alapján lehet kézi huzalozást megvalósítani.

Ennél a listánál sincs egy oldalon több mint 50 sor, de az egy jelhez tartozó pontpárok mindig ugyanarra az oldalra kerülnek.

Egy új feladat eredményei új oldalon kezdődnek.

Ha a jel csak egy pontot érint, akkor a pont koordinátái mellé a THE ISOLATED ELEMENT üzenetet írja a gép.

- Az eredményeknek a post-processor programok adatait képező része diszk file-ra is kerül (dsi=9).

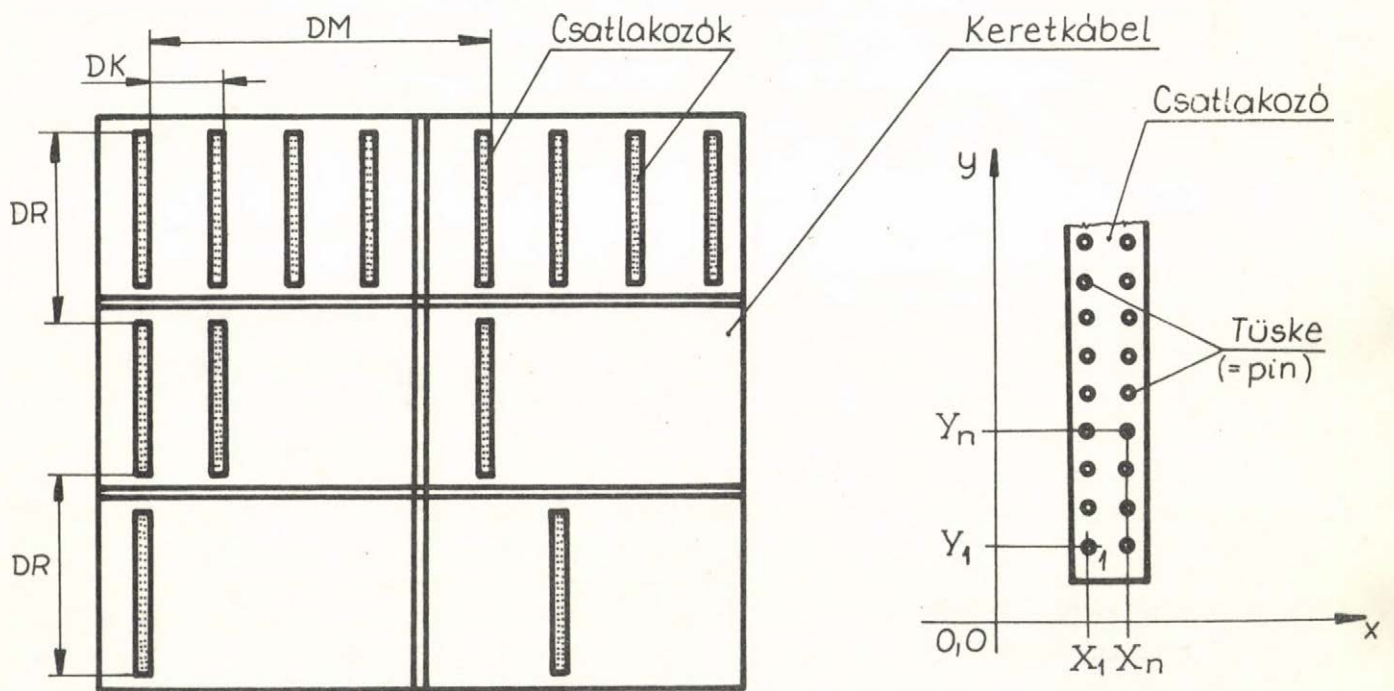
A diszkre írás formátuma a következő:

(5X,5HWORK=,A8,32X,6HFRAME=,I2,2X,2HN=,I2/3X,7HSIGNAL=,7A8)  
(6X,2(4X,1HR,I2,3H M,I2,3H K,I2,3H P,I2),7X,I4,4X,I2)

Ez a második formátum módosul az összeköttetlen pontok esetén:

(10X,1HR,I2,3H M,I2,3H K,I2,3H P,I2,19X,20HTHE ISOLATED ELEMENT)

- A programfutás végén, az utolsó összeköttetés után a program egy "end of file" jelet ír a diszkre (dsi=9)
- A különböző távolságok definícióját a 3. ábra adja meg. Az ábrán  
DK=Csatlakozó távolság (két szomszédos csatlakozó közti távolság)  
DM=Modultávolság (mint DK-nál)  
DR=Fióktávolság (mint DK-nál)  
X1=Egy csatlakozó 1. tuskéjének x koordinátája  
Y1=Egy csatlakozó 1. tuskéjének y koordinátája  
Xn=Egy csatlakozó n. tuskéjének x koordinátája  
Yn=Egy csatlakozó n. tuskéjének y koordinátája



3. ábra



- Ha a 2. típusú kártyán bármilyen adat hiányzik, akkor az előző feladat megfelelő adata érvényes, ha nincs ilyen, akkor hibaüzenet fog megjelenni.
- Jelenleg a 4. típusú adat a dsi=41 diszk file-ról, míg a többi adatkártyáról kerül a gépbe.

### Hibaüzenetek

| Üzenet                       | Jelentés  |
|------------------------------|---|
| NO INPUT DATA                | bemenő adatok hiánya  |
| SORRY, NO PARAMETER CARDS    | a 2. vagy 3. adattípus hiányzik   |
| SORRY, TOO MANY PINS         | a 3. típusú adat több, mint 100 rekordot tartalmaz  |
| SORRY, NO PINS PARAMETER     | a 3. típusú adatok hiánya   |
| SORRY, TOO MANY LENGTH-CLASS | az 1. típusú adat több rekordból áll, mint 100  |
| SORRY, NO LENGTH CLASS       | az 1. típusú adatok hiánya  |
| NO DATA FOR THIS WORK        | egy feladat adatainak hiánya  |
| UNDEF PIN                    | a 4. típusú adatok közt olyan tűskeszám van, amelyik nincs definiálva a 3. típusban. Ilyenkor a legmagasabb számú definiált tűskeszám kerül felhasználásra. |

- Az utolsó üzenet esetén a számítás zavartalanul folyik tovább, míg a többi hiba fatális, tehát befejeződik a programfutás az addig kiszámított eredmények kiírása után.
- A program *sikeres futását* egy:  
END OF CONEX üzenet jelzi.

### A bemenő adatok rendezése

Amennyiben szükség van rá a huzalozó program futása előtt a CDC 3300 Master operációs rendszerében működő SORT rutin használatával adatrendezést hajthatunk végre. Mind a négy adattípusra vonatkozik a rendezés, aminek két célja lehet:

- Ha az egyes feladatok és esetleg a jelek adatrekordjai nem rendezettek, akkor a rendezés egyrészt feladatonként, másrészt a feladatokon belül jelenként választja szét az adatokat úgy, ahogy az a program számára szükséges.
- A 2. és 3. típusú adatok (geometria) ugyanazt a feladatnevet kell, hogy tartalmazzák, mint a 4. típusúak. Ezért, ha az előbbieket még nem az utóbiakkal együtt szerepelnek, akkor a rendezés elvégzi ezt az összeválogatást.

A rendezés *prioritási sorrendje* (az alacsonyabb számúnak nagyobb a prioritása).

| Prioritás | Adat                        | Pozíció | Hossz | Tipus         |
|-----------|-----------------------------|---------|-------|---------------|
| 1         | feladat                     | 1–6     | 6     | alfanumerikus |
| 2         | keret                       | 63–64   | 2     | numerikus     |
| 3         | jel                         | 7–60    | 54    | alfanumerikus |
| 4         | tüskeazono-<br>sító v. fiók | 66–70   | 5     | alfanumerikus |
| 5         | modulszám                   | 71–72   | 2     | alfanumerikus |
| 6         | csatlakozó<br>szám          | 75–76   | 2     | numerikus     |
| 7         | tüskeszám                   | 79–80   | 2     | numerikus     |

## MEGJEGYZÉS

A 4. prioritás osztályban a tüske azonosító a 3. adattípusra vonatkozik. Az osztály típusa azért alfanumerikus, mert ha fiókra vonatkozik, akkor a 66, 69 és 70 pozícióban konstans van.

## A program részei és szubrutinjai

|               |  |
|---------------|--|
| PROGRAM TRANS | Beolvassa az összes adatot a kártyaolvasóból és a megfelelő file-okról és diszk file-ra írja (dsi=20). |
| PROGRAM CONEX | Huzalozó program, minimális fa számítással.  |

## Szubrutinok a CONEX programban

|        |   |
|--------|---|
| INGRP  | Beolvassa és rendezi a különböző típusú adatokat, hívja az XY rutint és jelenként kiírja a dsi=24 munka file-ra a jel összes adatát, előkészíti a CONEX főprogramot az ághosszak kiszámítására. |
| CRSRT  | A CONEX-ben kiszámított $N(N-1)/2$ ághossz nagyság szerinti rendezését végzi.   |
| OUTGRP | Minimális fa kiszámítás, felhasználható vezetékkategóriák kiválasztása, eredmények kiírása sornyomatóra és diszk file-ra.   |
| LAB    | Fejlécek kiírása, új lap kezdés, sorszámlálás kezdés.   |
| XY     | Az x és y irányú távolságok kiszámítása az ághosszak számításához.  |



## MEGJEGYZÉSEK

- A két program kapcsolódását és a szubrutinok kapcsolódását a CONEX programon belül a 4. és az 5. ábra mutatja.
- A programok nyelve: FORTRAN IV., de mivel a CDC 3300/MASTER DECODE könyvtári rutinját és esetleg a SORT rutint is használja jelen formájában csak a CDC 3300-on futtatható.
- Szükséges perifériák:

|                            |          |
|----------------------------|----------|
| Kártyaolvasó               | (dsi=60) |
| Sornyomtató                | (dsi=61) |
| 8. Diszk: adatbemenet      | (dsi=41) |
| az összes adat             | (dsi=20) |
| eredmények                 | (dsi=9)  |
| lefordított program        | (dsi=33) |
| Mágnesszalag a rendezéshez | (dsi=21) |

Ezekon kívül jelenleg a következő dsi-jű file-ok is sornyomtatóként deklarálandók, ha az összes eredményre szükségünk van: 2, 3, 4, 6.

Ha ezeket elhagyjuk, akkor az esetleges hibaüzeneteken kívül csak az összekötendő pontpárok (vezeték hossz + kategória) és a kategória darab-összhossz táblázat jelenik meg a sornyomtatón.

- Ha a CONEX az elhelyező programmal egy job-ban fut, akkor a \$FILE,6=OUT kártya nem tehető a job-ba, mert a dsi=6 file-t az elhelyező program munka-file-ként használja.
- Memória igény: 60 negyed lap  
(30k szó x 24 bit)
- A program futásideje egy kb. 1500 összekötést tartalmazó 40 lapból álló keretre 3–5 perc.

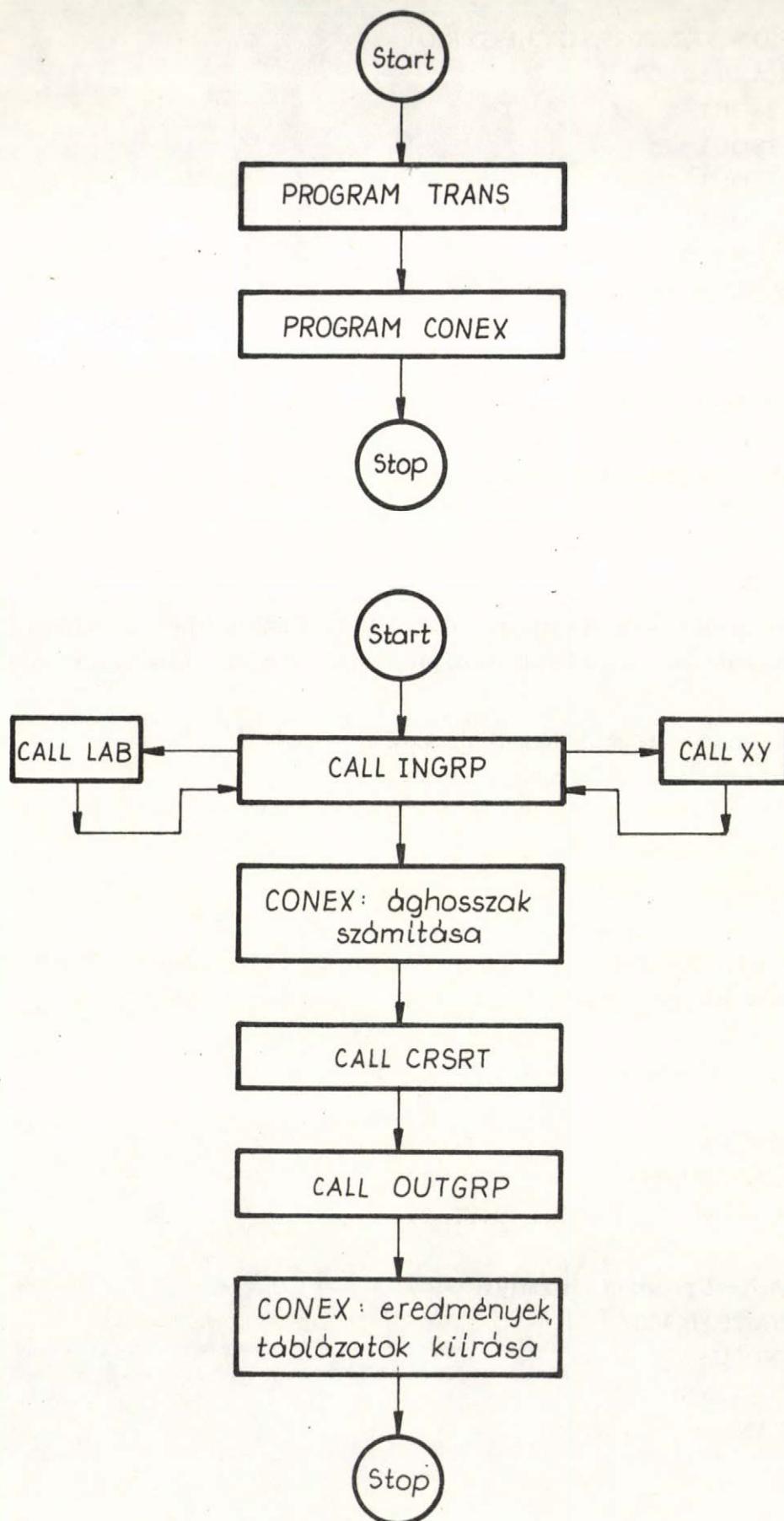
## Futtatási utasítás

Jelenleg a bemenő adatok közül az 1., 2. és 3. típusú lyukkártyáról, a 4. típus diszk file-ról (dsi=41) kerül a gépbe. Ha ezen módosítani akarunk, akkor csak a néhány soros TRANS programot kell átalakítani.

Egy programfutáshoz, ahol az adatok rendezésére is szükség van, a következőképpen kell a dekket összeállítani:

```
$JOB,szám,név,idő,sorok száma,,, megjegyzés
$SCHED,854=1,CORE=60,SCR=60,604=1
$*DEF(O,W,41,222025,KGY4,1,GYR,O)
$*DEF(O,W,9,222025,KGY5,1,GYR,O)
```





4. 5. ábra

```
*DEF(O,W,33,222025,KGY3,1,GYR,O)
$SOCR(A,20,80,300)
$FILE,2=OUT
$FILE,3=OUT
$FILE,4=OUT
$FILE,6=OUT
$FTNU(L,X,S,D)
```

TRANS program kártyái

```
$TRS,LGO
```

adatkártyák

```
*DEF(U,W,21,604,B,,,O)
$SORT
```

A rendezés paraméter kártyái (5 db)

```
$REWIND(21)
```

```
$X,33
```

EOJ kártya

- Ha a programot a CONEX kártyáival akarjuk futtatni annak újra fordításával a dsi=33-as file-ra vagy anélkül, akkor az elhelyező programnál ismertettel megegyező módon kell eljárni.
- Ha nem akarjuk a rendezést használni, akkor a \$SCHED kártyán a 604=1 nem szükséges és a \$\*DEF(U,W,21,604...), a \$SORT és az utána lévő 5 paraméterkártya helyére a \$FILE,21=20 kártya kerül.
- Ha egy job-ban akarjuk futtatni az elhelyező programot és a huzalozó programot, akkor a következő kártyacsomag összeállítása szükséges (feltételezzük, hogy nincs szükség az összes eredményre, ezért a dsi=2,3,4 és 6 OUT file-ok elmaradnak, tehát ezek az eredmények munka file-okra kerülnek, ahonnan szükség esetén visszanyerhetők):

```
$JOB,...
```

```
$SCHED,CORE=80,CLASS=B,SCR=120,854=1,604=1
```

```
$FILE,7=INP
```

```
$FILE,5=OUT
```

```
$SOCR(A,41,80,300)
```

```
*DEF(O,W,34,222025,KERT,1,GYR,O)
```

```
$X,34
```

Elhelyező program adatkártyái

```
$SOCR(A,20,80,300)
```

```
$REWIND(41)
```

```
$FTNU(L,X,D,S)
```

TRANS program kártyái

\$TRS,LGO

1., 2. és 3. típusú adatkártyák

\$\*DEF(U,W,21,604,B,,,O)

\$SORT

Rendezés paraméter kártyái

\$REWIND(21)

\$\*DEF(O,W,9,222025,KG5,1,GYR,O)

\$\*DEF(O,W,33,222025,KG3,1,GYR,O)

\$X,33

EOJ kártya

Látható, hogy a korábbi \$\*DEF(..41..) helyett \$SOCR(..41..) kártyával nyitottuk meg a dsi=41 file-ot. Ennek oka az, hogy a két program közvetlen egymás utáni futása miatt nincs szükség az átmenő adatok tárolására.

## MEGJEGYZÉSEK

- A CONEX program utolsó adatkártyája után egy olyan kártyát kell elhelyezni, amelynek mind a 80 pozícióján I betű van lyukasztva.
- A \$\*DEF file-ok helyett \$SOCR vagy külön nem deklarált munka file-ok is használhatók adattárolásra és átvitelre, ill. a munka file-ok (pl. 24) helyett is lehet \$\*DEF vagy \$SOCR file-t használni.
- Az eredményeket tartalmazó dsi=9 file-ra minden feladat után a következő formátummal még rekordokat ír ki a program (2X,I2//2X,I6), ahol az I2 és I6 formátumú egész számok értéke = 1;
- Ha a programfutás sikeresen befejeződik és az END OF CONEX üzenet megjelenik a sornyomtatón, akkor a dsi=9 file-ra a 99-es számot írja a program a következő formátummal: (62X,I2/). Ezek a záró kiírások a dsi=9 file-ra a post-processor programok miatt kellenek.
- Mellékelten megadjuk a \$SORT könyvtári rutin paraméter kártyáit.  
(Megjegyezzük, hogy a CDC 3300 legutóbbi átalakításai miatt a paraméter kártyák bizonyos módosítása válhat szükségessé.)



|    | Pozíció:               |    |  |    |    |
|----|------------------------|----|--|----|----|
|    | 1                      | 12 | 23   | 33 | 72 |
| 1. | 01105500N              |    | 11000600012100020063110056000711000500662100020071 |    |    |
| 2. | 0221000200752100020079 |    |  |    |    |
| 3. | 1 DS0080 UF            |    | 20   |    |    |
| 4. | 1 T00080 UF            |    | CM21   | x  |    |
| 5. | 9ENDMSS                |    |  |    |    |

A SORT rutin paraméter kártyái:

### 3. POST-PROCESSOR PROGRAM A TSK HUZALOZÓ FÉLAUTOMATÁHOZ

#### A program neve: POSTWI

A program a japán gyártmányú (TSK=Tokyo Seimitsu Co., Ltd.) S-WM-1A jelű félautomata hátaphuzalozó berendezés [1] vezérlőszalagját állítja elő a CONEX program eredményének felhasználásával.

A vezérlő lyukszalagot a huzalozó félautomatához kapcsolt N-PC-101A jelű NC vezérlő egység lyukszalagolvasója olvassa be. A lyukszalag szabvány NC vezérlő szalag és EIA RS-244-A kódban kell lyukasztani.

A lyukszalagon lévő információ hatására az NC egység ad működési utasítást a huzalozó és a huzaltároló egység részére, oly módon, hogy a huzalozó egység xy hídján lévő mutató beáll afölé a pont fölé, ahol a huzalozást végre kell hajtani, egyértelműen meghatározza a wire-wrap pisztoly helyét és amennyiben huzalkezdésről van szó, kijelöli a megfelelő kategóriájú huzalt a huzaltárolóban. A gép kezelője kiszedi az előkészített huzalt a tárolóból, a pisztoly segítségével huzaloz a megfelelő tűskére, majd gombnyomással továbblépteti az olvasót. (A huzaltároló, huzalozó és NC egység látható balról jobbra a 6. ábrán.) A gép egy max. 500x500 mm-es kereten képes dolgozni.

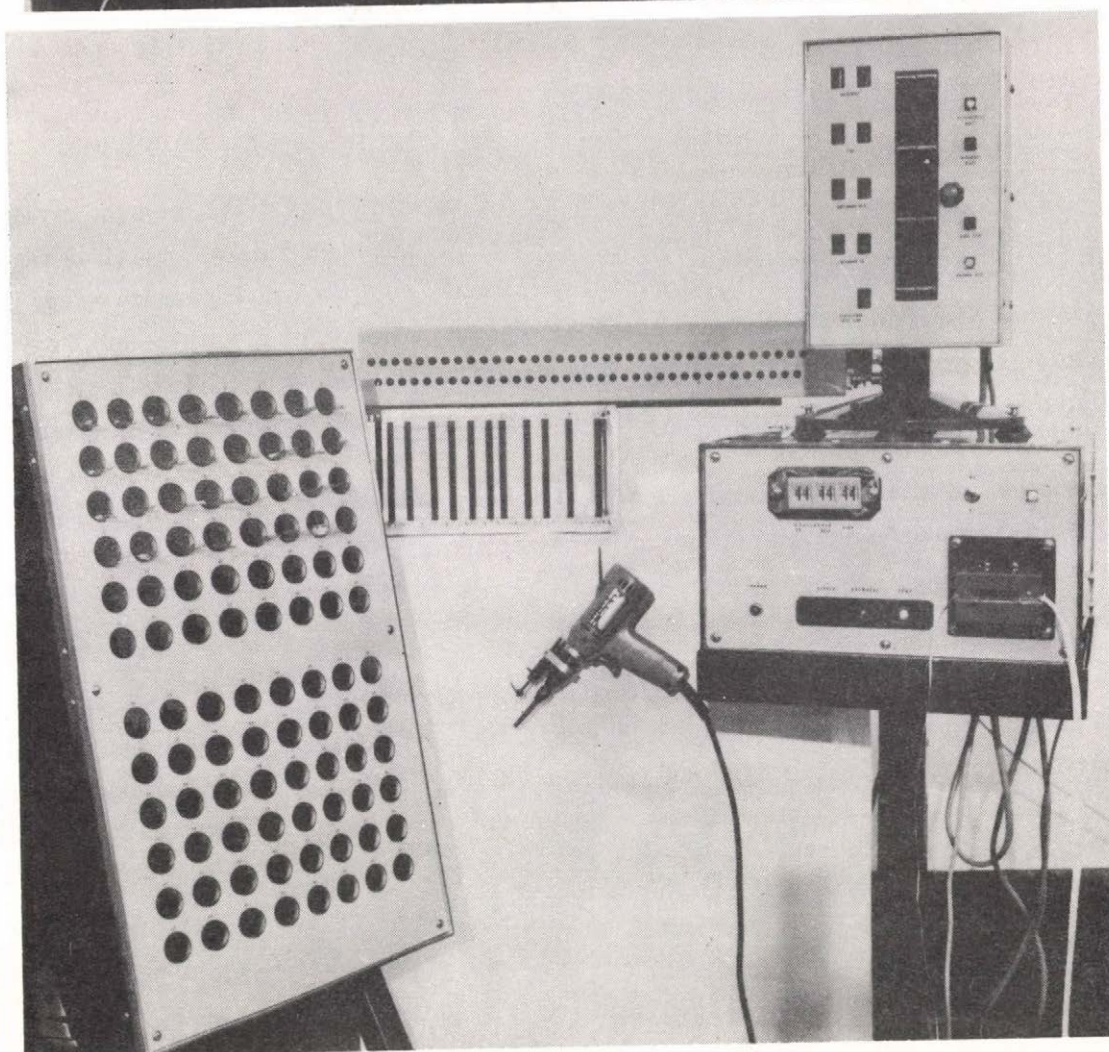
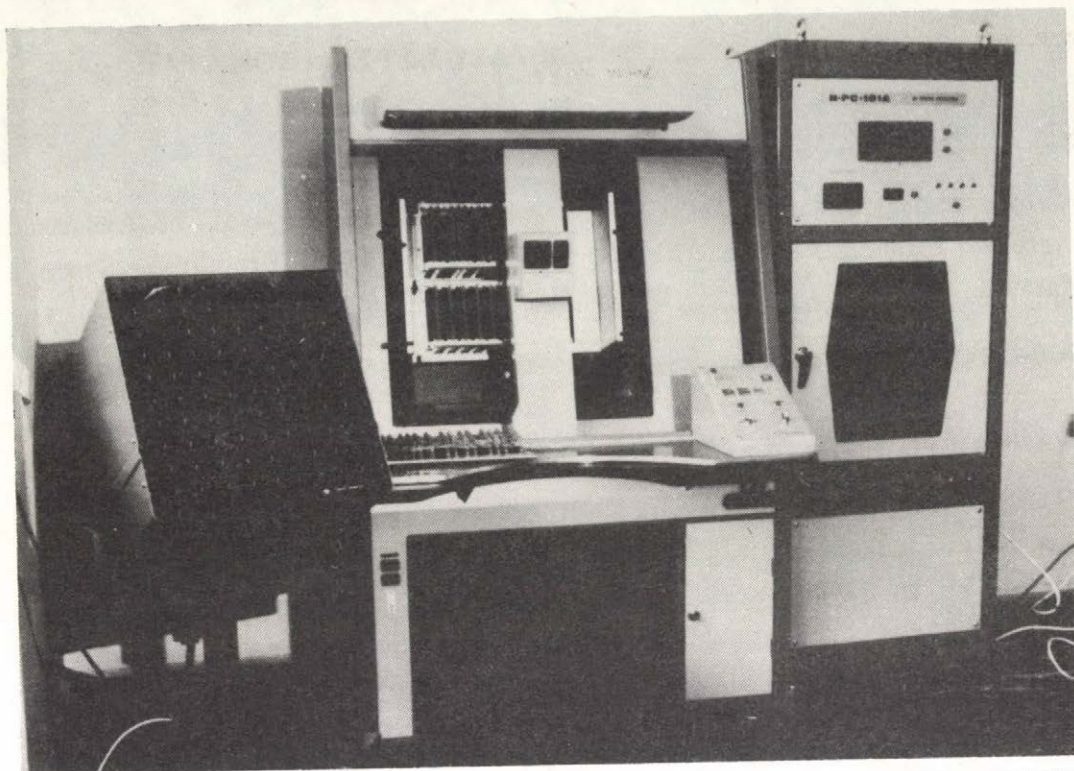
#### A program működése

A post-processor program lyukkártyáról vagy más 80 oszlopos formátumú bemenő egységről beolvassa a keret és a csatlakozók alapvető geometriai adatait, majd a dsi=9 jelű diszk file-ról a CONEX program eredményeként létrejött összekötendő pontpárokat. Egyszerre az egy jelhez tartozó összes összeköttetés beolvasása, majd feldolgozása történik meg, oly módon, hogy a huzalozás – kényelmi okokból – mindig balról jobbra történjék. Eredményként a program a sornyomtatón kiadja, hogy mit olvasott be és mit lyukaszt lyukszalagra. Egyidejűleg vezérlő lyukszalagot készít az automata számára. Az egy jelhez tartozó összes összeköttetés feldolgozása után a program rátér a következő csoportra, egész addig ismételve, amíg adatot talál a dsi=9 diszk file-on. (Az alap geometriai adatokat természetesen csak egyszer olvassa be.)

#### Adatelőkészítés

Mivel a diszk file-on lévő összekötési listát az előző programfutás állítja elő és annak formátumát a CONEX program leírásakor megadtuk, most csak a lyukkártyára lyukasztandó adatokat kell ismertetnünk. Ezek 2. csoportjaként viszont a CONEX 3. típusú bemenő adatai használhatók fel. Tehát a POSTWI program futásához tulajdonképpen csak egy adatkártya lelyukasztása szükséges.





6. 7. ábra



## 1. Alapvető geometriai adatok

Formátum: (2X,5I6)

Tartalom: IX,IY,KKR,KKM,KKK

|     |  |
|-----|--|
| IX  | A keret és a huzalozó automata nullpontjainak x irányú távolsága |
| IY  | A keret és a huzalozó automata nullpontjainak y irányú távolsága |
| KKR | Sortávolság (mint DR a 3. ábrán)                                 |
| KKM | Modultávolság (mint DM a 3. ábrán)                               |
| KKK | Csatlakozó távolság (mint DK a 3. ábrán)                         |

## 2. Csatlakozó tűske térképe

Formátum: (20X,2(2X,I7))

Tartalom: PINS(I,J),J=1,2

PINS(I,1): x koordináta

PINS(I,2): y koordináta

A beolvasás I szerinti növekvő egész számokkal történik, ahol I 1-től növekszik és a pin azonosító számát jelenti.

## MEGJEGYZÉS

- Az összes távolság 0.1 mm-es egységekben értendő.
- A program nyelve: FORTRAN IV., de felhasználjuk a CDC 3300/MASTER PTWRITE könyvtári rutinját lyukszalag lyukasztás céljából.
- A program helyfoglalása: 39 negyed lap (20K szó x 24 bit)
- Szükséges perifériák:

|                       |          |
|-----------------------|----------|
| Kártyaolvasó          | (dsi=5)  |
| Sornyomtató           | (dsi=6)  |
| Lyukszalaglyukasztó   | (dsi=10) |
| 8. diszk: bemenő adat | (dsi=9)  |
| lefordított program   | (dsi=31) |
- A program futásideje 1–2 perc egy normális feladatra, a futás költsége (KVAZ idő) a lyukszalaglyukasztó miatt ennél több.

## Futtatási utasítás

Ha a CONEX program sikeresen lefutott, akkor a következőképpen lehet a futtatáshoz a kártyacsomagot összeállítani:

\$JOB, szám, név, idő, sorok száma,,, megjegyzés  
\$SCHED,CORE=40,CLASS=B,SCR=5,3692=1,854=1  
\$\*DEF(O,W,9,222025,KGY5,1,GYR,O)  
\$\*DEF(O,W,31,222025,KGY1,1,GYR,O)  
\$\*DEF(U,W,10,3692)  
\$FILE,6=OUT  
\$FILE,5=INP  
\$X,31

Adatkártyák

EOJ kártya

- A programnak a programkártyákkal történő futtatásához a \$X,31 kártya helyett a következők szükségesek:

\$FTNU(X,L,S,D)

Programkártyák

\$X,LGO

- Amennyiben az összes adatot lyukkártyáról akarjuk beolvasni, akkor a \$\*DEF(O,W,9,...) kártya helyett a \$FILE,9=INP kártyára van szükség.
- Lehetőség van arra is, hogy a program a CONEX programmal egy job-ban futva a huzalozás számítása után azonnal elkészítse az automata vezérlő szalagját.
- Több feladat egymásutáni futtatása is lehetséges, ha az összes feladat ugyanazokat a geometriai alapadatokat használja. Minden feladat megkezdésekor a sornyomtatón a

DATA K                   üzenet és befejezésekor a

DATA K READY   üzenet jelenik meg, ahol K a feladat sorszámát (1-től növekvő egész) jelenti.

Ha az összes feladat futása sikeresen befejeződik, akkor a program a STOP 0002 üzenettel áll le.

#### 4. POST-PROCESSOR PROGRAM A MANU-WRAP BERENDEZÉSHEZ

A program neve: MOTHER

A program az MTA SzTAKI Digitális Technika Osztályán kifejlesztett [2, 3] Manu-Wrap nevű hátlaphuzalozó berendezés vezérlő szalagját állítja elő, bemenő adatként a CONEX programnak a dsi=9 diszk file-ra kiírt eredményeit használva.

A program működése

- A program jelenkénti csoportokban beolvasott adatokat feldolgozva ISO 7 kódban lyukasztott vezérlő szalagot készít a Manu-Wrap számára.

A lyukszalagolvasó által beolvasott vezérlő szalag a berendezés szám- és lámpa-kijelzőit működteti. Így kijelöli, hogy a huzaltároló egységből melyik kategóriájú vezetékre van szükség, a gépkezelő látja, hogy a keret melyik fiókjában, melyik csatlakozó, melyik tűskéjén kell huzalozni. Információt kap arra nézve is, hogy egy vezeték elejét vagy végét kell bekötni, valamint arra, hogy az adott tűskén hányadik kötés elhelyezése következik.

- A POSTWI nevű programhoz hasonlóan ez a program is képes egy futás folyamán több feladatot megoldani, a feladat megoldásának kezdetekor a

DATA K és befejezésekor a

DATA K READY üzenetet adva a sornyomtatón.

- Sikeres futás után a program a STOP 0002 üzenettel áll le.
- A program nyelve, memória és időigénye a POSTWI programéval megegyező.
- A program ugyanazokat a perifériális egységeket használja, mint a POSTWI és az egyes perifériák dsi-je is ugyanaz.

Az egyetlen eltérés az, hogy a program a dsi=32 file-ra van lefordítva.

**Futtatási utasítás**

A programhoz adatkártyák lyukasztása nem szükséges, de a program egy korábbi sikeres CONEX futást feltételez.

A futtatáshoz a következő kártyacsomagot kell összeállítani:

```
$JOB, szám, név, idő, sorok száma,,, megjegyzés
$$SCHED,CORE=40,CLASS=B,SCR=5,3692=1,854=1
$*DEF(O,W,9,222025,KGY5,1,GYR,O)
$*DEF(O,W,32,222025,KGY2,1,GYR,O)
$*DEF(U,W,10,3692)
$FILE,5=INP
$FILE,6=OUT
$X,32
EOJ kártya
```



## 5. POST-PROCESSOR PROGRAM A TSK HUZALOZÓ FÉLAUTOMATÁHOZ

A program neve: WIRECONN

A program célja:

A program a TSK huzalozó félautomatához készít vezérlőszalagot, az előzőektől eltérő bemenő adatrendszer felhasználásával.

- Abban az esetben célszerű ezt a programot felhasználni, amikor a tervezésnek a Bevezetésben leírt 1. és 2. része már valamilyen módon elkészült és a már kész huzalozási terv alapján akarjuk az automata vezérlő szalagját elkészíteni.
- A program lehetőséget ad különböző típusú (max. 10 fajta) csatlakozók definiálására, amelyek tetszőleges pozícióban helyezkedhetnek el a kereten. Természetesen minden egyes csatlakozó tipushoz meg kell adni annak tűske-térképét. Az egyes csatlakozókat ezután azonosítójuk, típusuk és elhelyezkedésük határozza meg.
- Lehetőség van a programban dupla és tripla sodrott vezetékek felhasználására. Ezeket az összekötési lista megadásakor D ill. T betű jelzi. (Természetesen T jelzésű összekötésből 3, D jelzésűből 2 kell, hogy egymás után szerepeljen.) Az egyszerű összeköttetés jelzésére az N betűt használjuk.
- A program tehát bemenő adatként a csatlakozóknak a kereten való elhelyezkedését, az egyes csatlakozó típusok tűske-térképeit, a vezetékkategóriákat és egy összekötendő pontpárokat tartalmazó összekötési listát kap. Ez a lista megadja, hogy mely csatlakozó, mely pontja, melyik csatlakozó melyik pontjával és milyen típusú vezetékkel (N, D vagy T) van összekötve.

### Adatelőkészítés

1. *kártya* üres, ha rack bal alsó sarkát tekintjük 0 pontnak, ha nem, akkor 0.01 mm-ben meg kell adnunk a gépi 0-tól számított x és y távolság értékét, 2 db 5 jegyű számmal.  
A beolvasás formátuma: (2I5)
  2. *kártya* a csatlakozó típusok számát jelzi  
a változó neve: ICSSZ  
a beolvasás formátuma: (I5)
  3. *kártyától* kezdődik az *I. kártyacsoport*. Ezek a kártyák tartalmazzák a *csatlakozók* azonosítóit és elhelyezkedését.  
A beolvasás formátuma: (A4,4X,4(A3,1X,2(I5,1X)))
- 1–4 pozícióba kötelezően CONN szó kerül  
5–8 pozíció üres: megjegyzésnek használható  
9–11 (ill. 25–27, 41–43, 57–59) 3 karakter (betű vagy szám), ami a csatlakozó azonosítója

- 12 (ill. 28, 44, 60) vessző vagy más elválasztójel
  - 13–17 (ill. 29–33, 45–49, 61–65) pozíciókba 5 jegyű szám kerül a csatlakozó egy pontjának (javasolt az egyik tűske ill. pin) x koordináta értéke 0.01 mm-ben
  - 18 (ill. 34, 50, 66) vessző vagy más elválasztójel
  - 19–23 (ill. 35–39, 51–55, 67–71) pozíciókba 5 jegyű szám kerül, ami a csatlakozó ugyanazon pontjának y koordináta értéke, 0.01 mm-ben
  - 24 (ill. 40, 56, 72) vessző vagy más elválasztójel
  - 73–80 üres vagy megjegyzésnek használható.
- A felsorolás folyamatosan történik. Egy adatkártyán 4 csatlakozó adható meg. Az utolsó csatlakozó koordinátáinak leírása után a soron következő helyre a név helyett szóköz kerül (ill. üresen marad), az x–y koordináta értékek helyett pedig 999,999 -et kell írni (ezt záró utasításnak nevezzük).
  - A CONN kártya csoportok száma egyezik ICSSZ-el, amit a 2. kártyán adtunk meg. Minden csoport tartalmazza a záró utasítást is. Egy csoporton belül csak azonos típusú csatlakozók szerepelhetnek.
  - Így a csatlakozók az 1. típusba tartoznak, amíg a gép be nem olvassa az első záró utasítást, majd az 1. és 2. záró utasítás között definiált csatlakozók a 2. típusba tartoznak és így tovább.

## II. Kártyacsoport

Az egyes csatlakozó típusok tűske–térképeit adja meg ugyanolyan sorrendben, ahogy az egyes típusokat az I. kártyacsoport definiálta. Az egyes tűskék x és y koordináta értékeit kell megadni 0.01 mm-ben ugyanahhoz a referenciaponthoz képest, mint amelyik az egyes csatlakozók elhelyezkedését határozza meg. A tűskék azonosítószámai a megadás sorrendjében 1-től növekvő egész számok minden egyes csatlakozó típusra.

A beolvasás formátuma: (A4,4X,12(I5,1X))

- 1–4 karakter kötelezően PINS
- 9–13 (ill. 21–25, 33–37, 45–49, 57–61, 69–73) 5 jegyű szám, ami egy tűskének (pin) az X koordináta értéke 0.01 mm-ben.
- 14 (ill. 26, 38, 50, 62, 74) vessző vagy más elválasztójel
- 15–19 (ill. 27–31, 39–43, 51–55, 63–67, 75–79) 5 jegyű szám, ami egy tűskének az Y koordináta értéke 0.01 mm-ben
- 20 (ill. 32, 44, 56, 68, 80) vessző vagy más elválasztójel

1 tűskéhez 2 szomszédos szám tartozik. Az első az x, a második az y koordináta értéke. Egy adatkártyán 12 tűske definiálható. Az utolsó számpár után következő koordináta értékek helyére 999,999 kerül, ami azt jelzi, hogy egy pin–térkép beolvasása megtörtént.

Egy csatlakozónak legfeljebb 100 tűskéje (pin) lehet.



Itt is ugyanúgy, mint a csatlakozóknál ICSSZ-szer kell szerepelni pin-térképnek, ill. a hozzá tartozó vége jelzésnek, azaz a 999,999-nek.

### III. Kártyacsoport

A vezetékek *hossz és minőség szerinti* választékát tartalmazza mm-ben. Egy adatkártyán 12 vezetékkategória adható meg.

A beolvasás formátuma: (A4,4X,12(A1,1X,I3,1X))

- 1–4 karakter kötelezően a WIRE szót tartalmazza
  - 5–8 üres vagy megjegyzésre használható
  - 9 (ill. 15, 21, 27, 33, 39, 45, 51, 57, 63, 69, 75) N, D vagy T betű, ami azt jelzi, hogy az utána következő számmal megadott hosszúságú vezeték egyszerű, duplán vagy triplán sodrott. (Az N, D és T betű másfajta minőségjelzésre is használható.)
  - 10 (ill. 16, 22, 28, 34, 40, 46, 52, 58, 64, 70, 76) üres vagy elválasztójel
  - 11–13 (ill. 17–19, 23–25, 29–31, 35–37, 41–43, 47–49, 53–55, 59–61, 65–67, 71–73, 77–79) 3 jegyű egész szám – a huzalkategória hossza mm-ben. Az utolsó érték után 999-et kell írni, ami a vezetékkategória lista végét jelzi.
  - 14 (ill. 20, 26, 32, 38, 44, 50, 56, 62, 68, 74, 80) üres vagy elválasztójelként (pl. vessző) használható.
- A hossz kategóriák szonosítói a beolvasás sorrendjében 1-től (max. 100-ig) növekvő egész számok. A megfelelő huzalhosszú vezeték a hozzá tartozó sorszámú tároló egységbe kerül.

### IV. Kártyacsoport: az összekötési lista

A beolvasás formátuma: (5(2(A3,1X,I2,1X),A1,1X))

A program kötéspárokat olvas be, a hozzá tartozó minőség jelzéssel. Ez a jelzés

- N – normál (1 kötéspár tartozik hozzá)
- D – dupla (2 kötéspár tartozik hozzá)
- T – tripla (3 kötéspár tartozik hozzá)

Egy sorba, azaz egy kártyára 5 összekötés kerül. Az összekötési lista folyamatosan, minden kártyát (sort) kitöltve irandó le. Az utolsó összekötés után a csatlakozónevek helyét üresen hagyjuk, a pin sorszáma helyett pedig 99,99-eket írunk, a minőség jelzés helyére V betűt. Ez jelzi a programnak az egész összekötési lista végét.

- 1–3 (ill. 17–19, 33–35, 49, 51, 65–67) 3 karakter, a csatlakozó neve, aminek szerepelnie kell a konnektor listában is. Ez vonatkozik az összekötési listában előforduló összes csatlakozónévre. Ennek valamelyik pinjére, amit a következő szám ad meg, kerül a vezeték eleje.
- 4 (ill. 20, 36, 52, 68) vessző vagy más elválasztójel



- 5–6 (ill. 21–22, 37–38, 53–54, 69–70). Az előző csatlakozó megfelelő sorszámu pinje.  
Ez egy 2 jegyű egész szám.
- 7 (ill. 23, 39, 55, 71) vessző vagy más elválasztójel
- 8–10 (ill. 24–26, 40–42, 56–58, 72–74) 3 karakter, a csatlakozó neve, aminek valamelyik pinjére, amit a következő szám ad meg, kerül a vezeték vége
- 11 (ill. 27, 43, 59, 75) vessző vagy más elválasztójel
- 12–13 (ill. 28–29, 44–45, 60–61, 76–77) az előző csatlakozó megfelelő pinjének a száma
- 14 (ill. 30, 46, 62, 78) vessző vagy más elválasztójel
- 15 (ill. 31, 47, 63, 79) a minőségjelzés ill vége jelzés betűjele. Ezek N, D, T vagy V lehetnek.
- 16 (ill. 32, 48, 64, 80) vessző vagy más elválasztójel.

## MEGJEGYZÉSEK

- A program eredményként a huzalozó automata vezérlőszalagját adja. A szalagon a huzalozást vezérlő utasítások sorrendje megegyezik az összekötési listán megadottal. Ezért pl. ha figyelembe akarjuk venni, hogy kényelmesebb, ha a gép minden összekötést balról jobbra valósít meg, akkor az összekötési listát ennek megfelelően kell elkészíteni.
- A futás minden eredményét ill. a beolvasott adatokat is kinyomtatja a sornymatató. Ha a futás során hiba történik, erről a sornymatató jelzést ad, de a programfutás tovább folyik. A hibás adatok feldolgozása nem kerül lyukszalagra.
- A futás végén táblázatot kapunk, amely azt tartalmazza, hogy melyik huzalkategória milyen típusú és hosszú vezetékeket tartalmaz és hogy abból hány darabra van szükség. Ezenkívül a felhasznált összvezeték hossz értékét is kiírjuk a sornymatatóra.

*A program nyelve:* FORTRAN IV, de felhasználjuk a CDC 3300/MASTER PTWRITE rutinját is.

*Memóriaigény:* 64 negyed lap (32K x 24 bit)

*Szükséges perifériák:*

|                               |          |
|-------------------------------|----------|
| 8. diszk: lefordított program | (dsi=51) |
| Kártyaolvasó                  | (dsi=5)  |
| Sornymatató                   | (dsi=6)  |
| Lyukszalaglyukasztó           | (dsi=13) |

*A program futásideje:* a feladat méretétől függően 1–2 perc.

### Futtatási utasítás

Mivel a lefordított program a dsi= 51 diszk file-on van, a program futása a következő kártyacso-maggal történhet:

```
$JOB, szám, név, idő, sorszám,,, megjegyzés  
$SCHED,CORE= 64,CLASS=B,SCR= 10,3692= 1,854= 1  
$*DEF(O,W,51,222074,WIRP,01,,I)  
$*DEF(U,W,13,3692)  
$FILE,5=INP  
$FILE,6=OUT  
$X,51
```

Adatkártyák  
EOJ kártya

## 6. POST-PROCESSOR PROGRAM A MANU-WRAP BERENDEZÉSHEZ

A program neve: MANUAL

A program célja: A WIRECONN programhoz hasonlóan ez a program is akkor használható, ha a huzalozás tervezése már befejeződött. A program az összekötendő pontcsoportoknak már a pontpárookra felbontott összekötési listája alapján dolgozik és készít vezérlő lyukszalagot a MANU-WRAP berendezéshez.

### Adatelőkészítés

*1. Kártya beolvasása* FORMAT(4I3) *szerint* történik.

A beolvasott paraméterek:

- LL a fiókok száma
- LX az egy fiókban elhelyezkedő max. csatlakozószám
- LH ha LH=0, akkor a vezeték hosszát a két pont egyenes összekötése adja  
ha LH=1 vagy 0-tól különböző szám, akkor a vezeték hossza a két pont derékszögű összekötéséből adódó hosszérték
- LK ha LK=0, akkor a kötés száma az addig előfordult kötések összessége, ha 0-tól különböző érték, akkor a potenciálsikonkénti kötésszámot jelzi ki a berendezést vezérlő lyukszalag.

A 2. kártyától kezdődik az *I. kártyacsoport*. Ezek a kártyák tartalmazzák a csatlakozók azonosítóit és koordinátáit (elhelyezkedését).

Minden csatlakozó egyforma típusú kell, hogy legyen és mindegyikhez ugyanazt a referencia-pontot használjuk a pozíciók meghatározásakor (pl. az 1. pint). Egy adatkártyán 4 csatlakozó adható meg.

A beolvasás formátuma: (A4,4X,4(I4,1X,2(I5,1X)))

- 1–4 pozícióba kötelezően CONN szó kerül
- 5–8 pozícióba üres (megjegyzésnek használható)
- 9–12 (ill. 26–29, 43–46, 60–63) 4 jegyű szám, amely a csatlakozó neve. Ez a helyzetét is meghatározza, mivel az első két számjegy a fiók, a második kettő a csatlakozó száma
- 13 (ill. 30, 47, 64) pozícióba vessző vagy más elválasztójel kerül
- 14–18 (ill. 31–35, 48–52, 65–69) pozíciókba 5 jegyű szám kerül a csatlakozó egy pontjának (javasolt az egyik tűske ill. pin) x koordináta értéke 0.01 mm-ben
- 19 (ill. 36, 53, 70) vessző vagy más elválasztójel
- 20–24 (ill. 37–41, 54–58, 71–75) szintén egy 5 jegyű szám a csatlakozó ugyanazon pontjának y koordináta értéke kerül, ugyancsak 0.01 mm pontossággal



25 (ill. 49, 52, 76) oszlopba vessző vagy más elválasztójel.

A felsorolás folyamatosan történik. Az utolsó csatlakozó koordinátáinak leírása után a soron következő helyre a név helyett négy space (szóköz) kerül, a x–y koordináta értéke helyére pedig 999,999 értéket kell írni. A program ennek hatására olvassa a további kártyákat.

Az oszlopkijelző (fiókkijelző) használata esetén a csatlakozókat az oszlopkijelző osztásának (lámpái számozásának) figyelembevételével kell számozni.

## II. Kártyacsoport

A vezetékek hossz szerinti választékát tartalmazza mm-ben. Egy adatkártyán 18 vezetékkategória adható meg.

A beolvasás formátuma: (A4,4X,18(I3,1X))

- 1–4 karakter kötelezően a WIRE szót tartalmazza
- 5–8 üres vagy megjegyzésre használható (sorszám)
- 9–11 (ill. 13–15, 17–19, 21–23, 25–27, 29–31, 33–35, 37–39, 41–43, 45–47, 49–51, 53–55, 57–59, 61–63, 65–67, 69–71, 73–75, 77–79) 3 jegyű szám – a hosszúság mm-ben, és a hossz az összekötendő pontok távolsága mm-re kerekítve. Az utolsó érték után 999 értéket kell írni, ami jelzi a programnak, hogy vége a vezetékkategóriák listájának.
- 12 (ill. 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80) pozíciókba vessző vagy más elválasztójel kerül.

## III. Kártyacsoport

Egy csatlakozón belül a tűskék elhelyezkedését mutatja.

A kártyára az x–y koordináta értékeket kell megadni 0.01 mm pontossággal, ugyanahhoz a ponthoz képest, amelyhez a csatlakozókat definiáltuk. Egy kártyára 12 tűske kerül.

A beolvasás formátuma: (A4,4X,12(I5,1X))

- 1–4 karakter kötelezően PINS
  - 9–13 (ill. 21–23, 33–37, 45–49, 57–61, 69–73) 5 jegyű szám, ami egy tűskének (pin) az x koordináta értéke 0.01 mm-ben
  - 14 (ill. 26, 38, 50, 62, 74) vessző vagy más elválasztójel
  - 15–19 (ill. 27–31, 39–43, 51–55, 63–67, 75–79) 5 jegyű szám, ami egy tűskének az y koordináta értéke 0.01 mm-ben
  - 20 (ill. 32, 44, 56, 68, 80) vessző vagy más elválasztójel.
- 1 tűskéhez 2 szomszédos szám tartozik, az első az x, a második az y koordináta értéke. Az utolsó számpár után következő koordináta értékek helyére 999,999 kerül, ami jelzi a programnak, hogy a pin-térkép beolvasása megtörtént.

- A számpárokat a pin számozás sorrendjében kell leírni. (Ez tetszőleges, de természetesen az összekötési listán ennek megfelelően kell rá hivatkozni.)

#### IV. Kártycsoport: összekötési lista

A beolvasás formátuma: (4(2(I4,1X,I2,1X),A1,1X))

A program jelcsoportokat olvas be. Egy jelhez max. 50 összekötés tartozhat. Négy összekötés fér el egy kártyára. A jelcsoport végét üresen hagyott összekötés jelzi. Ha úgy fejeződött be a jelcsoport leírása, hogy az utolsó összekötés éppen kitölti a kártyát, akkor utána egy üres kártyát kell betenni, ha nem, akkor csak új kártyán kell kezdeni a következőt. Az egész összekötési lista végét az utolsó jelcsoport után betett külön kártya adja, ami a 17. pozícióban egy V betűt tartalmaz.

- 1–4 (ill. 19–22, 37–40, 55–58) a csatlakozó neve, ami 4 jegyű szám. A vezeték kezdete.
  - 5 (ill. 23, 41, 59) vessző vagy más elválasztójel
  - 6–7 (ill. 24–25, 42–43, 60–61) a pin száma, amire a kötés kezdete kerül
  - 8 (ill. 26, 44, 62) vessző vagy más elválasztójel
  - 9–12 (ill. 27–30, 45–48, 63–66) a csatlakozó neve, amire a kötés vége kerül
  - 13 (ill. 31, 49, 67) vessző vagy más elválasztójel
  - 14–15 (ill. 32–33, 50–51, 68–69) pin száma, amire a kötés vége kerül
  - 16 (ill. 34, 52, 70) vessző vagy más elválasztójel
  - 17 (ill. 35, 53, 71) N betű, vagy üresen marad, ha pedig vége az összekötési listának, akkor V betű
  - 18 (ill. 36, 54, 72) vessző vagy más elválasztójel.
- Eredményül egy vezérlő lyukszalagot és a sornyomtató által készített listát kapjuk.
  - A lyukszalag tartalmazza a vezeték kategóriákat a hozzá tartozó összekötésekkel, valamint egy ellenőrző lyukszalagot, ami abból áll, hogy az 1. csatlakozó 1. pinjétől az utolsó csatlakozó utolsó pinjéig kijelzi az egyes tűskéken levő kötések számát, ami nem haladhatja meg a 4-et.  
Ezt a huzalozás elkészülte után ellenőrzésre használhatjuk.
  - Ha a futás során bármilyen hibajelzés történik, az ellenőrző szalag lelyukasztására nem kerül sor.
  - A listán megtalálható az összes adat, a kötésenkénti hosszúságok mm-re kerekítve, egy táblázat a vezetékkategóriákról és a hozzájuk tartozó szükséges darabszámról, valamint a csatlakozók képe, ami számjegyekből áll, azaz minden pin helyén a rajta lévő kötések száma látható (az ellenőrző lyukszalag adatai).
  - Ha a keretből elmenő vezetékeket akarunk megadni, akkor ezeket a 10. fiókhoz kötöttnek kell feltételezni és az eredménynél ennek hatására a 0-ik tűske kijelzése fog megtörténni.



*A program nyelve, memória igénye, futásideje megegyezik a WIRECONN programéval.*

*A szükséges perifériák is megegyeznek, csak ez a program a dsi=53 file-ra van lefordítva.*

#### **Futtatási utasítás**

A futáshoz szükséges kártyacsomag adatai és a lefordított programot tartalmazó file-t megnyitó kártya kivételével azonos a WIRECONN-nál leírtakkal.

Igy a

\$\*DEF(O,W,51...) kártya helyett a

\$\*DEF(O,W,53,21C224,MANUAL,01,O) és a

\$X, 51 helyett a

\$X,53 kártyára és természetesen az aktuális adatokra van szükség.

– A lefordított program a 8. diszk egységen van.



## 7. POST-PROCESSOR PROGRAM A MANU-WRAP BERENDEZÉSHEZ

A program neve: MANUAL2

A program célja:

A program a MANUAL nevű programmal azonos célt szolgál és ugyanazoknak a bemenő adatoknak a feldolgozására alkalmas.

Eltérés mutatkozik a program eredményében. A MANUAL a huzalozás vezérlőszalagját az összekötési listának megfelelő sorrendben készítette el; ez a program viszont rendezett sorrendben készíti el a huzalozást.

Külön-külön történik az egyes fiókokon belüli huzalozás, majd a fiókok közötti összeköttetésekre és végül a keretből elmenő vezetékekre kerül sor. Az egyes fiókokon belül a huzalozás vezetékhosszak szerint rendezett, úgy, hogy előbb a rövidebb, majd a hosszabb vezetékek kerülnek bekötésre. Így a gép kezelője egyszerre kézbeveheti az egyes huzalkategóriákhoz tartozó összes vezetékét és nem kell a vezetéktárolóval törődnie egészen addig, amíg az eggyel hosszabb vezetékekre nincs szüksége.

A hossz szerinti és a fiókonkénti rendezés viszont azzal a hátránnyal jár, hogy megszűnik az egy potenciált alkotó vezetékek egymás utáni huzalozása, ami a berendezés visszakereső lehetősége által nyújtott előnyök elvesztését eredményezi.

### Futtatási utasítás

A program a következőképpen összeállított kártyacsomaggal futtatható:

```
$JOB, szám, név, idő, sorok száma,,, megjegyzés  
$SCHED,CORE=65,CLASS=B,SCR=20,3692=1,854=1  
$*DEF(U,W,13,3692)  
$*DEF(O,W,51,222074,SZM,01,,I)  
$FILE,5=INP  
$FILE,6=OUT  
$X,51
```

Adatkártyák

EOJ kártya

- A lefordított program a 8. diszk egységen található.
- A program munka file-ként 7 és 8 dsi-jű diszk egységet használ.

## ÖSSZEFOGLALÁS

A jelen dolgozatban ismertetett programok szerves részét képezik az MTA SzTAKI Digitális Osztályán készülő, a digitális berendezések számítógépes tervezését megvalósító programrendszernek.

A programokat Intézeten belüli és kívüli felhasználók is igénybe vehetik és szolgáltatás jelleggel használhatják. Már eddig is meglehetősen sokan (Intézeten belüliek és külsők egyaránt) használták a programokat ill. a programok által kiszolgált berendezéseket.

A programokkal kapcsolatos felhasználói véleményeket igyekszünk figyelembe venni és szükség esetén módosításokat hajtunk végre. Így jött létre pl. MANUAL2 elnevezésű program.

1974. márciusában módunkban állt az elhelyező és a minimális huzalhosszal huzalozó programokat külföldi eredményekkel is összehasonlítani. A Szófiában lévő CIIT intézettel való együttműködésünk keretein belül egy-egy feladat adatait kicseréltük és mindkét feladatot megoldotta mindkét fél. Egy 25 és egy 40 csatlakozót tartalmazó keretkábelevezést, amelyek kb. 500 ill. 1200 összekötést tartalmaznak összehasonlítottunk és az eredményeket minőségileg közel azonosnak találtuk. A jövőben még néhány további feladat hasonló összehasonlítására is sor fog kerülni.

Jelenleg nagyszámú példán vizsgáljuk az elhelyező és huzalozó program hatékonyságát. A vizsgálat egyrészt az elhelyező program egyes paramétereinek (JAVSZ, MUST, KX, KY) helyes kiválasztási módszerét kell, hogy meghatározza, másrészt statisztikai vizsgálatokat végzünk a megvalósított huzalozás vezeték hossz-eloszlásaira vonatkozóan. Ezen munkák eredményeit egy későbbi tanulmány fogja ismertetni.

## IRODALOMJEGYZÉK

- [1] Semi-Automatic Wiring Machine (S-WM-1A) and Numerical Control System (N-PC-101A) Specifications and Operation Manual.  
Tokyo Seimitsu Co., Ltd., Tokyo, Japan, 1972.
- [2] Dibuz Ágoston-Gáspár János-Várszegi Sándor: MANU-WRAP, MSI-TESTER és TESZTOMAT-C berendezések ismertetése.  
MTA SzTAKI tanulmányok 10/1973
- [3] Dibuz Ágoston: MANU-WRAP hátlaphuzalozó berendezés.  
MTA SzTAKI belső kiadvány, 1973.
- [4] M. Loberman-A. Weinberger: Formal Procedures for Connecting Terminals with a Minimum Total Wire Length.  
Journal of ACM 4,4 (1957) pp. 428-433.



A TANULMÁNYOK sorozatban eddig megjelentek:

- 1/1973 Pásztor Katalin: Módszerek Boole-függvények minimális vagy nem redundáns,  $\{\wedge, \vee, \neg\}$  vagy  $\{\text{NOR}\}$  vagy  $\{\text{NAND}\}$  bázisbeli, zárójeles vagy zárójel nélküli formuláinak előállítására
- 2/1973 Вашкеви Иштван: Расчленение многосвязных мышленных процессов с помощью вычислительной машины
- 3/1973 Ádám György: A számítógépipar helyzete 1972 második felében
- 4/1973 Bányász Csilla: Identification in the Presence of Drift
- 5/1973\* Gyürki J.—Laufer J.—Girnt M.—Somló J.: Optimalizáló adaptív szerszámgépirányítási rendszerek
- 6/1973 Szelke Erzsébet—Tóth Károly: Felhasználói Kézikönyv (USER MANUAL) a Folytonos Rendszerek Szimulációjára készült ANDISIM programnyelvhez
- 7/1973 Legendi Tamás: A CHANGE nyelv/multiprocesszor
- 8/1973 Klafszky Emil: Geometriai programozás és néhány alkalmazása
- 9/1973 R. Narasimhan: Picture Processing Using Pax
- 10/1973 Dibuz Ágoston—Gáspár János—Várszegi Sándor: MANU—WRAP hátlaphuzalozó. MSI—TESTER integrált áramköröket mérő, TESTOMAT—C logikai hálózatokat vizsgáló berendezések ismertetése
- 11/1973 Matolcsi Tamás: Az optimum-számítás egy új módszeréről
- 12/1973 Makroprocesszorok, programozási nyelvek. Cikkgyűjtemény az NJSzT és SzTAKI közös kiadásában. Szerkesztette: Legendi Tamás
- 13/1973 Jedlovsky Pál: Új módszer bonyolult rektifikáló oszlopok vegyész-mérnöki számítására
- 14/1973 Bakó András: MTA Kutatóintézeteinek bérszámfejtése számítógéppel
- 15/1973 Ádám György: Kelet–nyugati kapcsolatok a számítógépiparban
- 16/1973 Fidrich Ilona—Uzsóky Miklós: LIDI–72 Listakezelő rendszer a Digitális Osztályon, 1972. évi változat
- 17/1974 Gyürki József: Adaptív termelésprogramozó rendszer (APS) termelő műhelyek irányítására
- 18/1974 Pikler Gyula: MINI–Számítógépes interaktív alkatrészprogramíró rendszer NC szerszámgépek automatikus programozásához

\*-gal jelölt kivételével a TANULMÁNYOK megrendelhetők az Intézet Könyvtáránál (Budapest, I. Uri u. 49.)



- 19/1974 Gertler, J.—Sedlak, J.: Software for process control
- 20/1974 Vámos, T.—Vassy, Z.: Industrial Pattern Recognition Experiment—A Syntax Aided Approach
- 21/1974 A KGST I–15–1. témában folytatott szeminárium előadásai
- 22/1974 Arató, M.—Benczúr, A.—Krámli, A.—Pergel, J.: Part I. Stochastic Processes
- 23/1974 Benkó Sándor—Renner Gábor: Erősen telített mágneses körök számítógépes tervezési módszere

Jelen dolgozat a 4.2.4. "Logikai áramkörök  
számítógépes tervezése" c. intézeti alapku-  
tatási téma keretében készült.

---

Beérkezett: 1974. május 9.







